
**CYBER 96X MSL
MAINTENANCE SOFTWARE
REFERENCE MANUAL**

IFT3/P TEST DESCRIPTION

**CDC® COMPUTER SYSTEMS
CYBER 960, 962**

PREFACE

=====

This manual describes the CDC CYBER 960 and 962 Computer System's IFT3/P test. Procedures for using this test are provided in the CYBER 96X Test Procedures Reference manual.

NOTE

It is assumed and planned that the user of program descriptions will have an in-depth understanding of the processor logic; including familiarity with all levels of the Multi-Level Block Diagrams.

DISCLAIMER

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

60000284 A

IFT3/P - 1

REVISION RECORD	
REVISION	DESCRIPTION
A (07-11-88)	Manual released.

Publication No.
60000284

REVISION LETTERS I, O, Q, S, X, AND Z ARE NOT USED

1988

by Control Data Corporation
Printed in the United States of America

60000284 A

Address comments concerning
this manual to:

Control Data Corporation
Technology and Graphics Division
4201 Lexington Avenue North
St. Paul, Minnesota 55126-6198

or use Comment Sheet in
the back of this manual.

IFT3/P - 11

CONTENTS

=====	
IFT3P. INSTRUCTION FETCH TEST.....	1
General.....	1
Test Initialization.....	5
Repeat Iteration.....	8
Section Descriptions.....	7
Section 0 - Basic P Path and Parity.....	7
Section 0, Subsection 0.....	7
Section 0, Subsection 1.....	8
Section 0, Subsection 2.....	10
Section 0, Subsection 3.....	11
Section 1 - CM Load to IF.....	14
Section 1, Subsection 0.....	14
Section 2 - Test Jump Instruction Assembly Multiplexer.....	18
Section 2, Subsection 0.....	18
Section 2, Subsection 1.....	19
Section 2, Subsection 2.....	22
Section 2, Subsection 3.....	23
Section 3 - Test Parcel Select Multiplexers.....	28
Section 3, Subsection 0.....	28
Section 3, Subsection 1.....	28
Section 4 - P Register Incrementer.....	31
Section 4, Subsection 0.....	31
Section 4, Subsection 1.....	33
Section 5 - Test the Branch Control Logic.....	38
Section 5, Subsection 0.....	38
Section 5, Subsection 1.....	37
Section 6 - Branch Address Adder Input Multiplexers.....	40
Section 6, Subsection 0.....	40
Section 6, Subsection 1.....	42
Section 6, Subsection 2.....	43
Section 6, Subsection 3.....	45
Section 6, Subsection 4.....	47
Section 7 - Test Branch Address Adder.....	50
Section 7, Subsection 0.....	50
Section 8 - Test Address Out of Range (AOR).....	53
Section 8, Subsection 0.....	53
Section 8, Subsection 1.....	58
Section 9 - Miscellaneous Tests.....	58
Section 9, Subsection 0.....	58
Section 9, Subsection 1.....	61
Section 9, Subsection 2.....	63
Section 9, Subsection 3.....	65
Section 9, Subsection 4.....	68
Section 9, Subsection 5.....	71
Section 9, Subsection 6.....	73
Section 9, Subsection 7.....	75
Section 9, Subsection 8.....	78

APPENDIXES

A. GLOSSARY.....	A-1
B. IFT3/P TEST MICRANDS.....	B-1

INSTRUCTION FETCH TEST IFT3/P

GENERAL

IFT3/P detects errors in the Processor Instruction Fetch logic. It resides in and executes from a PP. The assumption has been made that the PP has been tested and found to be error free. In addition, it is assumed that the maintenance channel interface and the Processor control store logic are free of errors.

The Instruction Fetch Logic reads instructions from Cache/Central memory, assembles, decodes, buffers, and subsequently emits these instructions to the Control Store and Instruction Control Pipeline sections of the Instruction Unit.

In addition to pre-fetching sequential instruction words, Instruction Fetch (IF) pre-fetches instructions in the branch direction under the assumption that branch conditions will be met at execution time. If, at execution time, the branch conditions are not met, IF will discard its buffer and resume fetching at the instruction word following the branch instruction that was not taken (Unbranch).

If a branch instruction is detected in the input stream for which the branch target address cannot be computed by IF (Exchange Jump, computed Branch, etc.), instruction pre-fetching will halt until a new program address is received from Operand Issue (OPI) via the DAI register. Signals are also available from the Instruction Control Pipeline (ICP) to IF to halt instruction fetching and discard the instruction buffer during initialization and Micro Trap sequences.

Instruction Fetch consists of six major areas:

1. Instruction Assembly and Parcel Selection
2. Instruction First Level Decode
3. Instruction Formatting
4. Instruction Buffer
5. Branch Address Network
6. P Register

The Instruction Assembly and Parcel Selection logic receives the 64-bit CM word from OPI. This logic checks parity on the CM word and selects a parcel to be sent to the first level decode logic and the instruction formatting logic.

The First Level Decode logic uses the opcode bits of an instruction to address a memory containing decode information. This decode information is sent to the instruction formatting logic and to various control logic within IF.

The Instruction Formatting logic formats the instruction information, first level decode information and control information into a form suitable for entry into the Instruction Control Pipe and Control Store.

60000284 A

IFT3/P - 1

The Instruction Buffer logic controls buffering of instructions before issue to the pipe (ICP).

The Branch Address Network has the task of forming the address of the next CM word to be requested from Local Memory (LM). This request may come as the result of a branch or RNI sequence.

The P register holds the byte address of the instruction being parcelled. It is incremented when an instruction enters buffer rank 2 by an amount equal to the instruction length in bytes. It may be reloaded with the current contents of the Branch Address Adder Result register.

The test (IFT3/P) begins by putting the logic in an idle state and testing the PFS registers for parity errors in the Instruction Fetch (IF) logic. This should uncover any static (erroneous) error status.

Following this, the P register path from Operand Issue through Instruction Fetch, through the Instruction Control Pipe, to the Register File is tested. This is initiated by a Live Register Write to the P Register. The value placed in the Register File is compared with the value of P written. The PFS registers are monitored for any Instruction Fetch P register path parity errors. Operands are included here to test the capability of the parity checkers to detect parity errors.

The next step in the test verifies the data path from CM to the instruction assembly buffer in Instruction Fetch. A failure here could indicate one of several problems. First, an incorrect address may have been presented for the fetch. Second, a failure between Central Memory and Instruction Fetch may have blocked the data. Third, a failure in Instruction Fetch may have prevented the data from getting into the assembly register. Data with parity errors is used to ensure that data is reaching the assembly register.

Once the data path from CM is found to be error free, the test goes on to verify the inputs to the Instruction Assembly Hybrid Mux. The BDP input is ignored at this point because of the amount of logic that must function properly to set up a test for this input. The BDP input will be tested later in IFT3/P.

The operand portion of this Mux is verified by having the microcode set up to store the referenced microcode address to a Register File location. The contents of this Register File location are then compared with the Control Store address that was expected to be referenced. The PFS registers are monitored for any parity errors in the logic being tested.

The non-opcode portion of this Mux is tested by loading, via microcode, the JKQ (C180) or the K (C170) portion of the virtual instruction from the ICP. This is then compared with the actual virtual instruction presented.

The parcel select Muxes are tested next. There are two parcel select Muxes. One is for C180 instructions and one is for C170 instructions. Each Mux output is connected to a separate input of the Instruction Assembly Hybrid Mux. The Muxes are tested by presenting varied data to the different parcels. Microcode is used to record the Control Store address referenced. The non-opcode portion is verified using microcode to read that portion of the virtual instruction from the ICP.

60000284 A

IFT3/P - 2

The P-register incrementer is tested in two parts; an increment of +2 and an increment of +4. The test consists of allowing the P register to increment from a starting value and of verifying the increment result. This is done by setting microcode to load the second P value from the ICP. Since the increment result could be anything (if in error), the microcode is set up to load P on the second instruction (no matter what it is). This P value is then compared against the value expected. The PFS registers are monitored for parity errors.

The Branch Address Adder network is tested with the various control inputs and operand values. The address result and PFS registers are used to verify proper operation. Testing is broken down into several subsections to aid in isolation of any errors.

The Address out of Range detector, gating of the Parcel 3 Save Register and the BDP input to the Instruction Assembly Mux are tested as a conclusion to IF testing.

Throughout the program description, reference is made to Register File locations. These are usually abbreviated as RFxx, where xx equals the register file address in hexadecimal.

Logic Terminology:

The processor logic circuit terminology has been updated to match the terminology used in publications level 1 through 4 diagrams. Circuit terminology used in the IFT3/P test code documentation (PP code and processor microcode) has not been updated. When in doubt about which terminology is correct (ERS or test code), the ERS terminology should be assumed correct.

Test Status Word:

Status words of interest to the test have been consolidated into a Test Status Word. This word consists of status bits taken from PFS registers 82, 86, 87 and 88.

Byte 0

8 X	PFS 86(40)	C170 Odd RAM A PE
4 X	PFS 86(41)	C170 Even RAM A PE
2 X	PFS 86(42)	C180 RAM A PE
1 X	PFS 86(43)	Rank 12 P bits 32 through 39/68 PE
X 8	PFS 86(44)	Rank 12 P bits 40 through 47/69 PE
X 4	PFS 86(45)	Rank 12 OpCode bits 0 through 7/P PE
X 2	PFS 86(46)	Instr Mux bits 16 through 23/66 PE
X 1	PFS 86(47)	Instr Mux bits 3,12 through 15,24/64 PE

Byte 1

8 X	PFS 86(48)	C170 Odd RAM B PE
4 X	PFS 86(49)	C170 Even RAM B PE.
2 X	PFS 86(50)	C180 RAM B PE.
1 X	PFS 86(51)	Rank 12 P bits 48 through 55/70 PE
X 8	PFS 86(52)	Rank 12 P bits 56 through 63/71 PE
X 4	PFS 86(53)	Instr Mux bits 33 through 40/68 PE
X 2	PFS 86(54)	Instr Mux bits 25 through 32/67 PE
X 1	Not Used	-----

Byte 2

8 X	PFS 86(56)	Br Adrs A bits 0 through 7/32 PE
4 X	PFS 86(57)	Br Adrs A bits 8 through 15/33 PE
2 X	PFS 86(58)	Br Adrs A bits 16 through 23/34 PE
1 X	PFS 86(59)	Br Adrs A bits 24 through 31/35 PE
X 8	PFS 86(60)	Br Adrs B bits 8 through 15/33 PE
X 4	PFS 86(61)	Br Adrs B bits 16 through 23/34 PE
X 2	PFS 86(62)	Br Adrs B bits 24 through 31/35 PE
X 1	PFS 86(63)	PFS PAK3 Parity Error

Byte 3

8 X	PFS 87(00)	Br Addr Inp PE 0
4 X	PFS 87(01)	Br Addr Inp PE 1
2 X	PFS 87(02)	Br Addr Inp PE 2
1 X	PFS 87(03)	Br Addr Inp PE 3
X 8	PFS 87(04)	Br Adrs bits 32 through 39/68 PE
X 4	PFS 87(05)	Br Adrs bits 40 through 47/69 PE
X 2	PFS 87(06)	Br Adrs bits 48 through 55/70 PE
X 1	PFS 87(07)	Br Adrs bits 56 through 63/71 PE

Byte 4

8 X	PFS 87(08)	Instr BFR RK2 bits 0 through 7/32 PE
4 X	PFS 87(09)	Instr BFR RK2 bits 8 through 15/33 PE
2 X	PFS 87(10)	Instr BFR RK2 bits 16 through 23/34 PE
1 X	PFS 87(11)	Instr BFR RK2 bits 24 through 31/35 PE
X 8	PFS 87(12)	Instr BFR RK11 bits 0 through 7/32 PE
X 4	PFS 87(13)	Instr BFR RK11 bits 8 through 15/33 PE
X 2	PFS 87(14)	Instr BFR RK11 bits 16 through 23/34 PE
X 1	PFS 87(15)	Instr BFR RK11 bits 24 through 31/35 PE

Byte 5

8 X	PFS 88(00)	Parcel byte 0 PE
4 X	PFS 88(01)	Parcel byte 1 PE
2 X	PFS 88(02)	Parcel byte 2 PE
1 X	PFS 88(03)	Parcel byte 3 PE
X 8	PFS 88(04)	Parcel byte 4 PE
X 4	PFS 88(05)	Parcel byte 5 PE
X 2	PFS 88(06)	Parcel byte 6 PE
X 1	PFS 88(07)	Parcel byte 7 PE

Byte 6

8 X	PFS	88(08)	Parcel 3 byte 0 PE
4 X	PFS	88(09)	Parcel 3 byte 1 PE
2 X	PFS	88(10)	PMUX byte 0 PE
1 X	PFS	88(11)	PMUX byte 1 PE
X 8	PFS	88(12)	PMUX byte 2 PE
X 4	PFS	88(13)	PMUX byte 3 PE
X 2	SS	00(80)	Processor not Halted (0=Halted)
X 1	Not Used		*-----*

Byte 7

8 X	PFS	82(02)	Cache Adrs Rgtr byte 0 PE
4 X	PFS	82(03)	Cache Adrs Rgtr byte 1 PE
2 X	PFS	82(04)	Cache Adrs Rgtr byte 2 PE
1 X	PFS	82(05)	Cache Adrs Rgtr byte 3 PE
X 8	PFS	82(06)	Cache Adrs Rgtr byte 4 PE
X 4	PFS	82(07)	Cache Adrs Rgtr byte 5 PE
X 2	PFS	88(03)	Micrand Address PE
X 1	Not Used		*-----*

Microcode Sequences:

IFT3/P usually sets up a test such that any virtual op code (i.e., reference to CSA 100-3FF(HEX)) will reference a micrand that will load RF3 with a value equal to the address referenced and then branch to CSA 7A(HEX). CSA 7A(HEX) will contain a micrand branch to the micrand sequence desired for the executing subsection. Exceptions to this are documented in the applicable subsections.

Refer to Appendix B for the micrand sequences used by the diagnostic.

TEST INITIALIZATION

1. Set Real Memory Address (RMA) mode (bit 82 of Processor DEC).
2. Set disable PDM (47) but retain first failure capture enable (34) and the margin bits while clearing the remaining bits in the Processor DEC register.
3. Set Central Memory SECEDED Mode (Central Memory DEC).
4. Determine CM size from Central Memory options installed register.
5. Load the Control Store overlay. Upon completion of the loading, clear shadow memory enable by clearing bit 32 of the DEC register.
6. Load AC, ALN soft control. *(SIMILAR TO PRODUCT SET)*
7. Determine if an I2 or an I4 is being used by reading the IOU EID Register. Set a flag to 0 for an I2 or 1 for an I4, for use later.
8. Clear Processor Test Mode Register.
9. Set C180 mode.
10. Write control store addresses 100-3FF with a micrand that loads RF3 with a value equal to the current CSA and then branches to CSA 7A(HEX).

TEST INITIALIZATION (continued):

11. Set Processor to monitor mode.+
12. Set 170 monitor mode flag.+
13. Clear Processor trap enables.+
14. Clear user condition mask.+
15. Clear monitor condition mask.+
16. Clear page size mask.+
17. Clear error exit mask.+
18. Clear IOU test mode register.

+ Steps 9-15 are performed by microcode sequence 95.

REPEAT ITERATION

All Repeat Iteration (SRI) commands repeat to condition zero unless otherwise specified.

SECTION DESCRIPTIONS

SECTION 0 - BASIC P PATH AND PARITY

This section verifies that the Instruction Fetch logic can be idled with no errors reported. It also tests the basic P (CM address) path from OPI through the Instruction Fetch logic.

A micrand will be executed to Initialize IF. This will initiate a branch sequence in Instruction Fetch. This sequence will not be completed until a Live Register Write to P is executed. During this idle time, the PFS bits relating to IF errors should be clear.

After the Live Register Write, the P register value will be sent to LM and a virtual instruction will be read from Central Memory. Parity checkers on these paths should provide detection and isolation for any errors.

Section 0, Subsection 0

This subsection tests the effect of a Master Clear on Instruction Fetch. The Master Clear sets the MAC OFF-LINE signal which causes the P register to load zero. The Master Clear signal clears the instruction assembly register bytes 2 through 7, clears request and response control, and halts Instruction Fetch. A Write to the Instruction Fetch decode RAMS is performed to clear any parity errors in the instruction assembly register bytes 0 and 1, or in the RAMS. Any errors in bytes 2 through 7 of the instruction assembly are blocked until an LM response. At this point, the PFS register should be clear of any Instruction Fetch error reports. The PFS registers will be read to verify this.

A Start is then issued from MAC. This will drop the MAC OFF-LINE (MACOL) signal and start Control Store. Instruction Fetch should remain idle because no CM request has been generated. If IF starts parcelling instructions, RF0 will go nonzero. This will also be true if IB12 is detected (erroneously) as being valid.

Hardware Tested:

- IF error enables (initial check).
- Initial check on MAC controls to IF.
- Initial check on P-Mux selects and parity.

Micrands Used:

Number	Description
7A	Branch to CSA 50.
50	Write constant (AA) to RF9.
10	If IB12 valid, exit.

60000284 A

IFT3/P - 7

Initialization:

- Write zero to CM addresses 0 through 10(HEX).
- Write CSA 7A with a micrand branch to CSA 50.

Conditions:

- C.0
 - a. Issue Master Clear to Processor.
 - b. Clear Register File. Write zeros.
 - c. Write zero data to Decode RAMS.
 - d. Clear PFS registers.
 - e. Read PFS registers 82, 86, 87, 88.
 - f. Report test status word; error if nonzero.
- C.1
 - a. Read and report contents of RF9; error if nonzero.
- C.2
 - a. Write data 10(HEX) to the Micrand Address Register.
 - b. Issue Start from MAC [This clears the MAC OFF-LINE (MACOL) signal].
 - c. Delay 100 microsec (approx.)
 - d. Read PFS registers.
 - e. Report the test status word; error if nonzero.
- C.3
 - a. Read and report contents of RF9; error if nonzero.
- C.4
 - a. Read and report contents of the Processor Micrand Address Register. Report error if not equal to 1810 (HEX).
- C.5
 - a. Report PFS Register 87 bits 19-23 in bits 59-83; informational.

Section 0, Subsection 1

This subsection tests the Branch Indexed control logic with an INITIALIZE signal (INLIZE) from Control Store. The Initialize will cause IF to start a branch sequence. The INTERRUPT HOLD FF will be set, followed by the BRINT FF. This will cause the BRANCH INDEX FF to set. Memory request logic will be cleared and IF will be prepared to accept a new P register value from OPI. The sequence will wait at this point until a Live Register Write to the P register, which will bring up the branch indexed request signal (BRIREQ) from ICP.

After the Live Register Write, Instruction Fetch (IF) logic will generate a request to Local Memory (LM). A new virtual instruction will be brought up from the CM location specified by the P register value written. The virtual instruction will be decoded and the resulting code will be sent to Control Store to form the address of the micrand to be executed. The resulting address will be within the range of 100(HEX) to 3FF(HEX). With the set up used, the address of the initial micrand executed is inconsequential. The micrand will cause the P value in the instruction Control Pipe to be written to address 4 of the Register File (RF4). This should be the same as the value written to the P register in Instruction Fetch.

60000284 A

IFT3/P - 8

Micrands Used:

Number	Description
7A	Branch to CSA 55.
55	Load RF4 = P and halt.
51	Initialize IF and halt.
52	Initialize IF, write P-Fetch from RF0, write RF4 with REV-DATE, exit.

Hardware Tested:

- Initialize Signal, INLIZE, to Instruction Fetch.
- Basic Branch Indexed Sequence.
- Instruction Buffer Clear.
- Instruction Buffer Parity Check.
- IB12VE (Rank 12 valid) signal to Control Store (CS).

Initialization:

- Issue Processor Master Clear.
- Write CSA7A with micrand branch to CSA 55.

Conditions:

- C.0
- Clear Central Memory error logs.
 - Write data 0 to RF0.
 - Write data FF(HEX) to RF4.
 - Execute micrand 051 at CSA 51 (Initialize IF and halt).
 - Wait Processor halted (timed).
 - Clear PFS registers.
 - Read PFS registers.
 - Report Test Status Word; error if no Processor halted.
- C.1
- Execute micrand 052 at CS52 (Write RF0 to P).
 - Wait Processor halted (timed).
 - Read PFS registers.
 - Report test status word; error if nonzero.
- C.2
- Read and report contents of Central Memory UCEL1; error if nonzero.
- C.3
- Read and report contents of Central Memory UCEL2; error if nonzero.
- C.4
- Master clear Processor.
 - Read and report contents of RF4; error if nonzero.
- C.5
- Clear Processor errors.
 - Get and report test status word; error if nonzero.
- C.6
- Read and report contents of micrand address register; informational.

60000284 A

IFT3/P - 9

Conditions:

- C.7
- Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Section 0, Subsection 2

This subsection tests the P register entry from Operand Issue through the Branch Address Adder to the P register in Instruction Fetch (IF). Writing the P register will initiate an instruction fetch from memory. The decoded instruction, along with the P register, will be sent to the Instruction Control Pipe (ICP) for execution.

The test will read the P register from the pipe (ICP) and compare it with the value written. Several values will be used to prove that the P register can be loaded properly. The PFS registers will be monitored for any parity errors in the P register path.

Control store addresses 100 through 3FF will contain micrand branches to the micrand sequence which will read the current P value from ICP. This should eliminate the effect of an error in the instruction op code or micrand address formed.

Hardware Tested:

- Branch Address Adder (BAA) A input Mux leg to Operand Issue.
- Branch Address Adder A input Mux register and parity checker.
- Branch Address Adder parity checker (A input leg).
- Branch Address register and parity checker.
- P register Mux input from Branch Address register.
- P register and P register parity predictor.
- Instruction Buffer Rank 12 register and parity check (P bits).
- Path in Instruction Control Pipe to rank 22.
- P register (RMA) to LM (Parity check only).

Micrands Used:

Number	Description
7A	Branch to micrand 55.
51	Initialize IF and halt.
55	Read P register and halt (Load RF4 = P) (Write P = RF0).
52	Initialize IF, write P-fetch from RF0, exit.

P-Values Used (32-bit values) are hexadecimal:

0---0, F---F, A---A, 5---5, 8---8, C---C, E---E, 7---7, 1---

60000284 A

IFT3/P - 10

If quick look not selected, the following are also used.

00 -- 00(HEX)
01 -- 01(HEX)
02 -- 02(HEX)
.
.
.
FE -- FE(HEX)
FF -- FF(HEX)
000000FF(HEX)
0000FF00(HEX)
00FF0000(HEX)
FF000000(HEX)

Conditions:

- C.0
 - a. Execute micrand 51 (initialize IF).
 - b. Clear PFS registers.
 - c. Read PFS registers.
 - d. Report status word; error if P-Mux parity error.
- C.1
 - a. Write P value used to RF0.
 - b. Execute micrand 52 (Write P register, exit).
 - c. Wait Processor halted (timed).
 - d. Read RF4.
 - e. Report contents of RF4; error if not equal value written to the P register (P value written is displayed as expected).
- C.2
 - a. Read PFS registers.
 - b. Report status word; error if P-Mux parity error.
- C.3
 - a. Report the operand number.
- C.4
 - a. Read and report the contents of the micrand address register; informational.
- C.5
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of test operands; else, repeat subsection for next operand.

Section 0, Subsection 3

This subsection tests the ability of the parity checkers on the P register path to detect and report errors. P register data with parity errors will be loaded via the DAI bus. The PFS registers will be monitored to determine if the errors are detected. The following PFS bits are verified to ensure that they are set when expected to be set. No error report is generated if bits other than the expected bits are set, unless the expected bit is clear. An error is reported if bits are not clear after a Clear Errors function.

60000284 A

IFT3/P - 11

Register	Bits
82	4 through 7
85	0 through 3
86	43, 44, 51, 52, 56 through 59
87	0 through 7
88	10 through 13

The P-register data with errors will be stored in register file location 0 (RF0) prior to the Live Register Write (LRW) to P.

Hardware Tested:

- Branch Address Adder A input Mux parity checker.
- Branch Address Adder (BAA) parity checker.
- Branch Address register parity checker.
- Instruction Buffer Rank 12 register parity checker (P bits).
- Error lines to PFS register bits.
- PFS register bits.

Micrands Used:

Number	Description
51	Initialize IF and halt.
52	Initialize IF, write P-fetch from RF0, exit.
55	Read P register to RF4 and halt.
7A	Branch to micrand 55.

P-Values Used:

00000002
00000200
00020000
02000000

Initialization:

- The initialization is the same as for subsection 2 and should still be in effect.

60000284 A

IFT3/P - 12

Conditions:

- C.0 a. Execute micrand 051 (Initialize IF).
b. Clear PFS registers.
c. Read PFS registers.
d. Report status word; error if P path parity errors.
- C.1 a. Write data used; with parity errors to RF0.
b. Report data used; informational.
- C.2 a. Write zero to RF4.
b. Execute micrand 52 (RF0 to P, exit).
c. Wait Processor halted (timed).
d. Read RF4.
e. Report contents of RF4 as actual and starting P as expected; error if not equal to the value written to P register.
- C.3 a. Read PFS registers.
b. Report Status word; error if not P path parity errors expected for P value used.
- C.4 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.
- | P-value | Status Word Expected Hexadecimal |
|----------|----------------------------------|
| 00000002 | 0008 1011 0000 0404 |
| 00000200 | 0010 2022 0000 0808 |
| 00020000 | 0800 4044 0000 1010 |
| 02000000 | 1000 8088 0000 2030 |
- C.4 a. Read and report the contents of PFS register 85. Report an error if not equal as expected.
- C.5 a. Report operand number.
- C.6 a. Read and report the contents of the micrand address register; informational.
- C.7 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands; else, repeat subsection for next operand.

SECTION 1 - CM LOAD TO IF

This section tests the data path from CM to IF. Errors detected in this section have a high probability of being in logic external to IF. See the subsection documentation for additional explanation.

Section 1, Subsection 0

This subsection tests to see that the proper data is brought from central memory (CM) to the instruction assembly register. In testing for correct data from CM, the Assembly Register parity will be monitored. If this is correct there are only a few other errors possible: the wrong CM address may have been referenced, the data to Instruction Fetch (IF) was blocked elsewhere, or a failure in IF is preventing the gating of CM data into the Instruction Assembly register.

In order to provide the isolation necessary, operands with byte parity errors will be used first. This technique should provide a clear indication of any bad gating terms on the path being tested.

NOTE

Data with errors coming from memory will cause error log entries in the memory. These are used to verify the P Rgtr value sent to CM. Instruction Assembly parity errors that are out of place could be the result of not blocking parity errors after a master clear.

Hardware Tested:

- Instruction Assembly byte 0, 1 Mux.
- Instruction Assembly register and parity checkers.
- C180 parcel select Mux output.
- Bits 4 through 11, 43, 49 of Instruction Mux (IAHY) output.
- Instr Bfr RK2 (IB020A bits 18 through 22, IB020B bits 19 through 22, IB020L bits 0 through 35).
- Instr Bfr RK12 (XIB121 bits 16 through 40, 66 through 68).

Micrands Used:

Number	Description
51	Initialize IF and halt.
52	Initialize IF, Write P-Fetch from RF0, and exit.
78	Move value equal to CSA to RF5, Move K portion of operand to RF7, Move JKQ portion of operand to RF8, Move P from rank 50 of ICP to RF4 and halt.
7A	Branch to micrand 78.

Operands Used: Hexadecimal

Number	(64 bits)	Bytes in Error
01	FF0---0	All bytes
02	40010--0	0
03	80010--0	1
04	60010--0	2
05	50010--0	3
06	48010--0	4
07	44010--0	5
08	42010--0	6
09	41010--0	7
10	0100--00	
11	0----0	
12	F----F	
13	A----A	Quick Look
14	5----5	Without parity errors
15	01---01	
16	8C---8C	
17	73---73	
18	0101---01	
19	0202---02	
20	0303---03	
271	FDFD---FD	Full test (in addition to Quick Look patterns)(No parity errors)
272	FEFE---FE	
273	FFFF---FF	

Post Subsection Operations:

- Restore SECCED mode in Central Memory.
- Clear CM addresses 0 and 1.
- Execute micrand at CSA52 to flush parity errors.

SECTION 2 - TEST JUMP INSTRUCTION ASSEMBLY MULTIPLEXER

This section tests the INSTRUCTION ASSY MUX (IAHY). This Mux is in three parts. Bits 1, 4 through 11, 41 through 43, 49, 50 and BDP start (BDPST) are on a portion selected by C180 mode, C170 Odd address RAM, C170 even address RAM or C180 BDP. Bits 3, 12 through 15, 24, PARCEL 3 ASE (ASEEN) and SIZNA are selected by C180 mode, C170 mode or C180 BDP. Bits 16 through 40 are selected by C180 or C170 mode.

This Mux is used to assemble an instruction for execution. Its output is passed via intermediate buffering to the Instruction Control Pipe or the Control Store, as appropriate for execution.

Each leg, with the exception of the BDP legs, will be tested using several data patterns. The BDP leg will be tested after the parcelling Mux has been tested (See S09, SS03). The PFS registers will be monitored for any parity errors.

Section 2, Subsection 0

This subsection tests the multiplexing of C180 instructions in the INSTRUCTION MUX, IAHY. Inputs to this Mux are selected with the following conditions: C180 mode, C180 BDP, C170 mode, C170 mode and even RAM address, or C170 mode and odd RAM address. When testing the C180 input, the C170 decode RAMs will be set to ones. This should cause the Mux to output ones if a C170 input to Instruction Mux bits 5 through 11 is selected.

Note that this subsection tests a path from the Instruction Assembly Register, IAPR through IF to Control Store and the Instruction Control Pipe, ICP. An error anywhere along this path could appear to be an Instruction Mux failure.

Hardware Tested:

- INSTRUCTION MUX, IAHY.
- Instruction Buffer Rank 12, IB12.
- C180 Parcel Mux output.

Micrands Used:

Number	Description
51	Initialize IF and halt.
52	Move RF0 to IF P register and exit.
76	Move value equal to CSA to RF5, Move K portion of operand to RF7, Move JKQ portion of operand to RF8, Move P from rank 50 of ICP to RF4 and halt.
7A	Branch to micrand 76.

Data Used (Hexadecimal):

Operand	Expect C180 JKQ	Expect C170 K
0000	00 0000	00 0000
FFFF	FF FFFF	1F BFFF
AAAA	AA AAAA	09 2AAA
5555	55 5555	18 9555
7373	73 7373	13 9373
8C8C	8C 8C8C	01 0C8C

Initialization:

- Write data ones to C170 decode RAMs.
- Write data zeros to C180 decode RAMs.
- Master clear Processor.
- Write 800(HEX) to RF0. (Byte address.)

Conditions:

- C.0 a. Report operand number; information only.
- C.1 a. Write test operands to CM 100(HEX) and 101(HEX) (Word address).
b. Clear PFS registers.
c. Execute micrand 052 (Write RF0 to P, exit).
d. Wait Processor halted.
e. Read PFS registers 86, 87, 88 (GTS subroutine).
f. Report test status word; error if nonzero.
- C.2 a. Read and report contents of RF4; error if not equal 800(HEX).
- C.3 a. Read contents of RF5.
b. Report actual buffer equal contents of RF5; expected buffer equal micrand address formed by op code; error if not equal.
- C.4 a. Read RF6.
b. Report actual buffer equal contents of RF6; expected buffer equal bits 8 through 31 of test operand; error if not equal.
- C.5 a. Read RF7.
b. Report actual buffer equal contents of RF7; expected buffer equal output of instruction Mux read with C170K formatting; error if actual and expected are not equal.
- C.6 a. Read and report the contents of the micrand address register; informational.
- C.7 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational:

Exit subsection if end of operands; else repeat subsection for next operand.

60000284 A

IFT3/P - 19

Section 2, Subsection 1

This subsection tests the C170 EVEN RAM address leg of the op code portion of the INSTRUCTION MUX. This portion of the Mux will select one of four inputs (BDP/C180/C170 EVEN RAM/C170 ODD RAM). For this subsection the C170 RAM tied to the leg not under test will be set to ones. The C170 EVEN RAM will be set to the test data. The test data will be referenced with a FF(HEX) op code. This should cause ones to appear at the other inputs to the Mux when zeros are expected at the leg under test.

The following data table shows the relationship between the Hexadecimal data stored in the referenced Decode RAM and the Hexadecimal data presented to the four legs of the Mux. The CM data will be kept to OFF0 1FE0 3FC0 7F80(HEX) so that address FF(HEX) of the RAM will always be referenced. Correct operation will be verified by comparing the micrand address sent to CST against that expected.

C170 Op bits 26 through 20	DATA TO INSTR MUX (Bits 4 through 11)				CS ADDRESS (MAR) EXP
	BDP	C180	C170 ODD	C170 EVEN	
00	20	FF	7F	00	1800
5A	20	FF	7F	5A	1B5A
25	20	FF	7F	25	1B25
33	20	FF	7F	33	1B33
66	20	FF	7F	66	1B66
4C	20	FF	7F	4C	1B4C
7F	20	FF	7F	7F	1B7F

Hardware Tested:

- C170 Parcel select Mux output.
- INSTRUCTION MUX (IAHY bits 3 through 11).
- INSTRUCTION BUFFER RANK 2 (IB020A bits 18 through 21, 32; IB020B bits 19 through 22).
- Rank 12 op code bits 0 through 7, 65.
- Modified op code (Control store address) lines to Control Store.
- Modified op code logic in Control Store.
- RA+FL register load.

60000284 A

IFT3/P - 20

Micrands Used:

Number	Description
51	Initialize IF and halt.
52	Initialize, move data from RF0 to IF P register, exit.
78	Move value equal to CSA to RF5, Move K portion of operand to RF7, Move JKQ portion of operand to RF8, Move P from rank 50 of ICP to RF4 and halt.
5C	Initialize IF and clear stream, Write RAC from RF1, Write RA+FL from RF2, Go to halt.
7A	Go to 78.

Initialization:

- Write 0FF0 1FE0 3FC0 7F80(HEX) to CM address 0 through 5.
- Clear Central Memory error logs.
- Write RAC=0 and RA+FL=FFFFFF(HEX).
- Write CSA 7A with a micrand branch to CSA 78.
- Set C170 mode.

Conditions:

- C.0
- a. Write test data to C170 Even Decode RAM.
 - b. Write data ones to C170 Odd Decode RAM.
 - c. Write ones to C180 Decode RAM (with parity errors).
 - d. Report operand number.
- C.1
- a. Clear PFS registers.
 - b. Execute micrand 052 (Initialize, Load P, and halt).
 - c. Wait Processor halted (timed).
 - d. Read PFS registers.
 - e. Report test status word; error if nonzero with exception of 180 RAM parity errors.
- C.2
- a. Read and report contents of RF5; error if not micrand address (C170 Even input) corresponding to test data. The expected address is reported as the expected value.
- C.3
- a. Read and report contents of micrand address register informational.
- C.4
- a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.
- Exit subsection if end of test data; else, repeat subsection for next test data.

Section 2, Subsection 2

This subsection tests the C170 ODD RAM address leg of the op code portion of the Instruction Mux. This portion of the Mux will select one of four inputs (BDP/C180/C170 Even RAM/C170 Odd RAM). For this subsection the C170 RAM tied to the leg not under test will be set to ones. The C170 Odd RAM will be set to the test data. The test data will be referenced with a FF op code. This should cause ones to appear at the other inputs to the Mux when zeros are expected at the leg under test.

The following data table shows the relationship between the Hexadecimal data stored in the referenced Decode RAM and the Hexadecimal data presented to the four legs of the Mux. The CM data will be kept to 0FF8 1FF0 3FE0 7FC0(HEX) so that address FF(HEX) of the RAM will always be referenced. Correct operation will be verified by comparing the micrand address sent to CS against that expected.

C170 op bits 26 through 20	DATA TO IAHY (Bits 4 through 11)				CS ADDRESS (MAR) EXP
	BDP	C180	C170 EVEN	C170 ODD	
00	20	FF	7F	00	1800
5A	20	FF	7F	5A	1B5A
25	20	FF	7F	25	1B25
33	20	FF	7F	33	1B33
68	20	FF	7F	68	1B68
4C	20	FF	7F	4C	1B4C
7F	20	FF	7F	7F	1B7F

Hardware Tested:

- INSTRUCTION MUX (IAHY) bits 3 through 11.

Micrands Used:

Number	Description
51	Initialize IF and halt.
52	Initialize move data from RF0 to IF P register, exit.
78	Write RF5 = RF3, Write RF7 with C170 K from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-current from ICP, Go to halt.
7A	Go to CSA78.

Initialization:

- Write 0FF8 1FF0 3FE0 7FC0 to CM address 0.
- Clear Central Memory error logs.

Conditions:

- C.0
- a. Write test data to C170 ODD DECODE RAM.
 - b. Write data ones to C170 EVEN DECODE RAM.
 - c. Write ones to C180 DECODE RAM.
 - d. Report operand number; informational.
- C.1
- a. Clear PFS registers.
 - b. Execute micrand 052 (Initialize, Load P, and halt).
 - c. Wait Processor halted (timed).
 - d. Read PFS registers.
 - e. Report test status word; error if nonzero.
- C.2
- a. Read and report contents of RF5; error if not micrand address (C170 Odd input) corresponding to test data. The expected address is reported as the expected value.
- C.3
- a. Read and report contents of micrand address register; informational.
- C.4
- a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.
- Exit subsection if end of test data; else, repeat subsection for next test data.

Section 2, Subsection 3

This subsection tests the C170 leg of bits 12 through 40 of the INSTRUCTION MUX. This portion of the Mux will select one of three inputs (C170/C180/BDP). The data input to the Mux will be selected and varied in an attempt to prove proper selection. The output will be verified using micrands to read the instruction (excluding the op code) in both C170 and C180 format. The IF Decode RAMs will be cleared to prevent unexpected control variations. The following table shows the relationship between the data and the instruction portion as read from the Instruction Control Pipe.

Operands Used (Hexadecimal):

CM Data	C170 iJK	C180 JKQ
0000 0000 0000 0000	00 0000	00 0000
FFFF FFFF FFFF FFFF	03 FFFF	77 FFFC
0555 4AAA 9555 2AAA	01 2AAA	52 4AA8
0AAA B555 8AAA D555	02 D555	25 5554
0262 64C4 C989 9313	01 9313	23 4C4C
0DCF 9B9F 373E 8E7C	02 8E7C	74 B9F0

Hardware Tested:

- INSTRUCTION MUX (IAHY) bits 13 through 40.
- Instr Bfr RK2 (IB02L bits 04 through 35).
- Instruction Buffer Rank 12 (XIB121 bits 16 through 40, 66 through 88).
- Instruction Buffer Rank 12 (XIB121 bits 16 through 40, 66 through 88).
- C170 Parcel Select Mux.
- Lines to Control Store.

Micrands Used:

Number	Description
52	Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
5D	Write RF7 with C170 K from ICP, Write RFXx with the contents of RF5, where xx is equal to 10(HEX) plus the value of the C170 i instruction portion, Write RF8 with C180 JKQ from ICP, Write RF4 with P-current from ICP, Go to halt.
7A	Go CSA 58.

Initialization:

- Write zeros to all of the decode RAMs.
- Write zero to RF0.
- Write CSA 7A with a micrand branch to CSA 58.

Conditions:

- C.0 a. Clear PFS registers.
 b. Preset RF5 with
 Byte 5 = Random data
 Byte 6 = C6(HEX)
 Byte 7 = Operand index
 c. Write operand to CM Word address 0.
 d. Execute micrand 052 (initialize IF, Load P, exit).
 e. Wait Processor halted (timed).
 f. Read PFS registers.
 g. Report test status word; error if nonzero.
- C.1 a. Read and report contents of RF6; error if not equal expected, see table of operands used.
- C.2 a. Read and report contents of RF7; error if not equal expected data, see table of operands used. Note that an error here could indicate an error in parcelling.
- C.3 a. Read and report contents of RF4; error if not equal to byte address in RF0; see expected data for this condition.
- C.4 a. Report operand number.
- C.5 a. Read and report contents of micrand address register; informational.
- C.6 a. Search register file addresses 10(HEX) through 17(HEX) for data equal to the contents of RF5. Report RF address if found; else report 18(HEX) (error if not equal as was expected).
- C.7 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands; else, repeat for next operand.

Operand Number	Expected Results
0	10
1	17
2	12
3	15
4	14
5	11

SECTION 3 - TEST PARCEL SELECT MULTIPLEXERS

This section tests the IF parcel select logic. This logic consists of two multiplexers; one for C170 instructions and one for C180 instructions. These are 30-bit and 32-bit multiplexers respectively. The Mux input is controlled by bits 61 and 62 of the IF P register. The multiplexers will be tested using various combinations of data in the different parcels.

Section 3, Subsection 0

This subsection tests the C180 PARCEL SELECT MUX. This Mux will select the instruction parcel to be executed from the word of virtual instructions read from central memory. The selection is controlled by bits 61 and 62 of the byte address (P register contents).

The CM word used will be set up with all ones in the parcels not being tested. The bits of the parcel being tested will be varied to ensure that all bits are working.

Micrands Used:

Number	Description
52	Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU-micrand, Write RF3 with REV-DATE from FU-micrand, Exit.
76	Write RF5 from RF3, Write RF7 with C170 K from ICP, Write RF6 with C180 JKQ from ICP, Write RF4 with P current from ICP, Go to halt.
7A	Go to CSA 76.

Parcel Data Used (Hexadecimal):

0000, FFFF, AAAA, 5555, 7373, 8C8C

CM Data Used (Hexadecimal):

XXXX XXXX FFFF FFFF
FFFF XXXX XXXX FFFF
FFFF FFFF XXXX XXXX
FFFF FFFF FFFF XXXX

NOTE

The parcel data indicated above will replace the XXXX as the parcels are tested. Parcel 0 of the following CM word will be used when testing parcel 3.

Initialization:

- Write CSA 7A with a micrand branch to CSA 76.
- Write ones to the C170 decode RAMs.
- Write zeros to the C180 decode RAMs.

Conditions:

- C.0
- a. Write CM byte Address to RF0.
 - b. Assemble CM word with parcel X and X+1 equal to parcel data. The rest of the word will be equal to ones.
 - c. Write CM word to CM Address 0.
 - d. Report parcel (24 and 25) and operand number (20 through 23); informational.
- C.1
- a. Clear PFS registers.
 - b. Execute micrand 052 (initialize and write P).
 - c. Wait Processor halted (timed).
 - d. Read RF4.
 - e. Report contents of RF4; error if not equal to byte address of instruction parcel being tested.
- C.2
- a. Read and report contents of RF5; error if not equal op code + 100(HEX) of instruction to be executed.
- C.3
- a. Read PFS registers.
 - b. Report Status word; error if instruction assembly parity errors.
- C.4
- a. Read RF8.
 - b. Report actual = contents of RF8.
Expected = JKQ portion of virtual instruction used; error if expected and actuals do not compare.
- C.5
- a. Read and report contents of micrand address register; informational.
- C.6
- a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of test operands; else, repeat subsection for all Parcel values. Repeat subsection for all parcels.

Section 3, Subsection 1

This subsection tests the C170 PARCEL SELECT MUX. This Mux will select the instruction parcel(s) to be executed from the CM word read. The selection is controlled by bits 61 and 62 of the instruction byte address (P register). Thirty bits are controlled with each select.

The CM word used will be set up with all ones in the parcels not under test. The bits of the parcel under test will be varied to ensure that all bits are working.

The C170K portion of the instruction will be read from the ICP and used to verify proper parcelling. The micrand sequence referenced will store the micrand address to RF5. This will be used to verify the op code portion of the instruction.

Hardware Tested:

- C170 Parcel Select Mux.
- P register bits to the C170 parcel Mux (bits 61, 62).

Micrands Used:

Number	Description
52	Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
5C	Initialize IF and clear stream, Write RAC from RF1, Write RA+FL from RF2, Go to halt.
76	Write RF5 from RF3, Write RF7 with C170 K from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-current from ICP, Go to halt.
7A	Go to CSA 76.

Instruction Data Used (Hexadecimal):

0--0, 3FFF FFFF, 2AAA AAAA, 1555 5555, 3373 7373, 0C8C 8C8C

CM Data Used (Hexadecimal):

Index	Data	
00	0000 0000 3FFF FFFF	****
01	0FFF FFFF FFFF FFFF	
02	0555 5555 BFFF FFFF	Test
03	0AAA AAAA 7FFF FFFF	Parcel 0
04	0CDC CDCD FFFF FFFF	
05	0323 2323 3FFF FFFF	****
10	0FFF E000 0000 7FFF	****
11	0FFF ----->F	
12	0FFF F555 5555 7FFF	Test
13	0FFF EAAA AAAA FFFF	Parcel 1
14	0FFF F9B9 B9B9 FFFF	
15	0FFF E848 4848 7FFF	****
20	0FFF FFFF C000 0000	****
21	0F ----->F	
22	0FFF FFFF EAAA AAAA	Test
23	0FFF FFFF D555 5555	Parcel 2
24	0FFF FFFF F373 7373	
25	0FFF FFFF CC8C 8C8C	****
30	0FFF FFFF FFFF 8000	****
31	0F ----->FF	
32	0FFF FFFF FFFF AAAA	Test
33	0FFF FFFF FFFF D555	Parcel 3
34	0FFF FFFF FFFF F373	
35	0FFF FFFF FFFF 8C8C	****

C170 RAM Data:

The CST entry bits of the C170 RAMs (First Level Inst Decoder) will be set to correspond to bits 0 through 6 of the selected virtual data. The intent here is to provide a Control Store address equivalent to bits 0 through 6 of the virtual data.

Initialization:

- Write the C170 Decode RAMs with bits 0 through 6 of the address in the Control Store entry field.
(Refer to RAM data table.)
- Set RAC=0 and RA+FL=FF FFFF(HEX).
- Set C170 mode.

Conditions:

- C.0 a. Report operand index (20 through 23) and parcel (24, 25).
- C.1 a. Write test word to CM.
b. Write byte address of instruction to be executed (Parcelled) to RF0.
c. Report CM word used; informational.
- C.2 a. Clear PFS registers.
b. Execute micrand 052 (write RF0 to P, exit).
c. Wait Processor halted (timed).
d. Read and report contents of RF4; error if not equal the byte address of the parcel being tested. The byte address is displayed as expected data.
- C.3 a. Read and report contents of RF5; error if not equal the expected micrand address. Expected micrand address is 300(HEX) + the op code + 1800(HEX). This is displayed as expected data.
- C.4 a. Read PFS registers.
b. Report test status word; error if nonzero.
- C.5 a. Read and report contents of RF7; error if not equal to C170 K portion of the instruction tested. The K portion should be equal to zero for parcel 3 operations.
- C.6 a. Read and report contents of micrand address register; informational.
- C.7 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of test operands; else, repeat for next operand.

SECTION 4 - P REGISTER INCREMENTER

This section tests the P register incrementer. Refer to the subsection documentation.

Section 4, Subsection 0

This subsection tests the P register incrementer. As instructions are parcelled out to the instruction buffer, the P register is incremented to maintain a pointer to the instruction being parcelled. The amount of the increment will be +2 if the instruction just parcelled is one parcel long. The increment will be +4 if the instruction is two parcels long. The instruction size is controlled by bit 0 of the C180 op code. This subsection will vary the initial P register value to test the P increment logic with an increment of +4. The micrand set up ensures that the second micrand executed halts the Processor.

This subsection will be run in C170 mode with the C170 RAM set to interpret every instruction as a two-parcel instruction.

Hardware Tested:

- P register increment and parity predict.
- Some of instruction buffers 2, 3, 10, and 11.

CM Words Used:

0-----0

P Register Values Used With +4 Increment (Hexadecimal):

- a- Walk a zero down from the top (with carry in).

```
7FFF FFFE
8FFF FFFE
DFFF FFFE
EDFF FFFE
|
|
|
FFFF FFFC
FFFF FFFE
```

P Register Values Used With +4 Increment (Hexadecimal):

- b- Walk a 0110 pattern down (with carry in).

```
8FFF FFFE
BFFF FFFE
D7FF FFFE
EDFF FFFE
|
|
|
FFFF FFB6
FFFF FFDA
FFFF FFF6
```

- c- Special (without carry in).

```
FFFF FFFC
FFFF FFF2
5555 5554
```

Micrands Used:

Number	Description
71	If DAI condition D, Write RF7 with C170K from ICP, Write RF4 with P-current from ICP, Go to halt. Else, Set DAI condition D, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU-micrand, Write RF3 with REV-DATE from FU-micrand, Exit.
7A	Go to CSA 71.

Initialization:

- Write CSA 7A with a micrand branch to CSA 71.
- Write C170 decode RAMs with Size equal to 30 bit.
- Clear CM 0 through 110(HEX).

Conditions:

- C.0
 - a. Write test address to RF0.
 - b. Report contents of RF0; informational.
- C.1
 - a. Execute micrand 075 (clear DAI condition bits, write RF0 to P).
 - b. Wait Processor halted.
 - c. Read and report RF4; error if not equal P+4 (condition 0 report plus 4).
- C.2
 - a. Read PFS registers.
 - b. Report test status word; error if P path parity errors.
- C.3
 - a. Report the operand number; informational.
- C.4
 - a. Report the contents of RF3; informational.
- C.5
 - a. Read and report contents of micrand address register; informational.
- C.6
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if last P value; else, repeat subsection for next P value.

Section 4, Subsection 1

This subsection tests the P register incrementer with an increment of +2.

The P register incrementer is used to advance the P register as instructions are parcelled out to the instruction buffer. The increment will be +4 if the instruction parcelled is two parcels (32 bits) long. The instruction size is defined by bit 0 of the instruction op code in C180 mode.

This subsection will execute single parcel instructions to keep the increment a +2. The P register value of the second instruction executed will be compared with the expected increment result to verify that the increment was correct.

Hardware Tested:

- P register incrementer.
- PPLUS 2 signal from the instruction Mux.

CM Data Used (virtual instruction): (Hexadecimal)

00-----00 (zero)

80000284 A

IFT3/P - 33

P Register Values Used:

```

0 --- 0
0 --- 02
0 --- 04
0 --- 06
0 --- 08
0 --- 0A
0 --- 0C
0 --- 0E
1FC
1FE

```

Micrands Used:

Number	Description
71	If DAI condition D, Write RF7 with C170K from ICP, Write RF4 with P-current from ICP, Go to halt. Else, Set DAI condition D, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU-micrand, Write RF3 with REV-DATE from FU-micrand, Exit.
7A	Go to CSA 71.

Initialization:

- Write CM data to test addresses.
- Write zero to C180 decode RAM.

Conditions:

- C.0
 - a. Write P value (byte address) to RF0.
 - b. Report contents of RF0; informational (this is initial P value).
- C.1
 - a. Execute micrand 075 (clear DAI condition Reg. write RF0 to P).
 - b. Wait Processor halted (timed).
 - c. Read and report contents of RF4; error if not equal P + 2.
- C.2
 - a. Read PFS registers.
 - b. Report test status word; error if P path parity errors.

80000284 A

IFT3/P - 34

Conditions:

- C.3 a. Report operand number.
- C.4 a. Read and report contents of micrand address register; informational.
- C.5 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if last P value; else, repeat subsection for next operand.

SECTION 5 - TEST THE BRANCH CONTROL LOGIC

This section tests the Branch control logic. Branches are initiated by C180 decode RAM bits 0 or 5, C170 decode RAM bits 0 or 6, or C170 decode RAM bit 11 and the Instr Stack Purge Enabled (IFPURG) signal from CST.

The Branch is aborted on receipt of an UNBR signal from ALN and a Rank 41 Enb1 Write 1 (ENUNBR) signal from ICP. Testing will use all of these signal paths. The number of input values used will be kept to a minimum, since the Branch Address Adder and input Multiplexers have not been tested. The UNBR signal will only be used in the zero state. The logic required to set the signal has not been tested at this point.

Section 5, Subsection 0

This subsection tests the Branch Indexed control logic using bit 42 of the instruction assembly Mux. Bit 42 comes from the first level decode RAMs (branch indexed bit). When set this bit will initiate a branch sequence which will transfer execution to a new instruction address. The new address will come from the DAI (OPI).

To test this, all words of the decode RAMs will be set up with these bits set. Control Store will be set up so that the second virtual instruction executed will load P to RF4 and halt Processor. If the branch executed as intended, the value loaded will equal the expected new address.

Hardware Tested:

- Instruction Assembly Mux bit 42 to Branch Control.
- Branch Control logic.
- C180 Decode RAM branch indexed bit to the Instruction Mux.

Micrands Used:

Number	Description
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7C	If DAI condition D, Write RF4 with P-current from ICP, Go to halt. Else, Write P-fetch from RF11, Set DAI condition D, Exit.
7A	Go to CSA7C.

Initialization:

- Write CSA 7A(HEX) with a micrand branch to 7C(HEX).
- Write C180 decode memory with the branch indexed bit set throughout.
- Set RF11=FFF(HEX).
- Write zero data to CM address 1FF(HEX) through 209(HEX).

Conditions:

- C.0
- a. Clear PFS registers.
 - b. Execute micrand 075 (clear DAI condition bits, RF0 to P, exit).
 - c. Wait Processor halted.
 - d. Read and report RF4; error if not equal to FFF(HEX).
- C.1
- a. Read PFS registers
 - b. Report status word; error if nonzero.
- C.2
- a. Read and report the contents of the micrand address register; error if not equal to 1803(HEX).
- C.3
- a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Section 5, Subsection 1

This subsection tests the branch control logic. Branch instructions will be executed with target addresses of parcels 0, 1, 2 and 3. The action taken when the branch target address is parcel 3 differs considerably from action taken with other target addresses. First, the CM word containing the target parcel must be loaded. Then, a request for the CM word following that must be requested in order to get the target parcel into the parcel 3 save register where it will be available to the parceling Mux.

The virtual instruction at the target address will read the current P register from ICP and store it in RF4. This will be compared with the expected (target) address. The instruction will also read the JKQ portion from ICP to RF6. This will be used to verify that the following cm location has been loaded in the condition where parcel 3 is the target address.

Hardware Tested:

- Branch control logic associated with parcel 3 target address.
- CM request logic.

Micrands Used:

Number	Description
51	Initialize IF, Go to halt.
59	If DAI condition A, Go to 59-1, Else, set DAI condition A, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7A	Go to CSA59.

CM Data Used (Hexadecimal):

CM data will be set up to execute branches from parcel 0 in CM address 0 to parcels in CM address 2. Parcel 0 of CM address 3 will be used when executing parcel 3 of the previous address. The following Hexadecimal data sets will be used.

Branch to Parcel 0

CM 00	8000 0008 F ---- F (64 bits)
CM 01	F ----- F
CM 02	8000 0001 F ---- F
CM 03	A ----- A

Branch to Parcel 1

CM 00	8000 0009 FFFF FFFF
CM 01	F ----- F
CM 02	FFFF 8000 0002 FFFF
CM 03	A ----- A

Branch to Parcel 2

CM 00	8000 000A FFFF FFFF
CM 01	F ----- F
CM 02	FFFF FFFF 8000 0003
CM 03	A ----- A

Branch to Parcel 3

```
CM 00 8000 000B FFFF FFFF
CM 01 F ----- F
CM 02 FFFF FFFF FFFF 8000
CM 03 0004 AAAA AAAA AAAA
```

Initialization:

- Execute micrand 051 (initialize IF).
- Clear PFS registers.
- Write CSA 7A(HEX) with micrand branch to 59(HEX).
- Set the branch indexed bit throughout the C180 decode RAM.

Conditions:

- C.0 a. Write data to CM address 0 through 3 (Word Address).
b. Report operand number; informational.
- C.1 a. Write zeros to RF0 and RF8.
b. Clear PFS registers and M3 error logs.
c. Execute micrand 075 (clear DAI conditions, write RF0 to P, exit).
d. Wait Processor halted (timed).
e. Read PFS registers.
f. Report test status word; error if nonzero.
- C.2 a. Read and report contents of RF4; error if not equal the branch target address.
- C.3 a. Read and report contents of RF8; error if not equal the JKQ of the virtual instruction at target parcel.
- C.4 a. Read and report the contents of micrand address register; informational.
- C.5 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Increment target address.

Exit Subsection if target address equal 8; else, repeat subsection for next target address.

Target Parcel	JKQ
0	1
1	2
2	3
3	4

SECTION 6 - BRANCH ADDRESS ADDER INPUT MULTIPLEXERS

This section tests the various inputs to the BRANCH ADDRESS ADDER (BAA). There are five inputs multiplexed into the A input of the adder. This is done in two levels. Inputs to the first level Mux (the one farthest from the adder) consist of: P register output bits 32 through 63, C170 reference address bits 0 through 26 and Instruction Buffer rank 2 bits 0 through 35. Inputs to the second level Mux (closest to the adder) consist of: the output of the first level Mux, New P bits 32 through 63 from OPI and ICP rank 50 P bits 0 through 31.

There are three inputs multiplexed into the B input of the adder. These consist of: the output of the literal generator, the C170 K times 4 from IB02 and the C180 Q times 2 from IB02.

The shortest path to the adder on the A input is the New P input from OPI through IF and out to Local Memory (LM). This has been tested in Section 0. The next shortest path is from rank 50 of the ICP. The more involved paths are the inputs from the P register, the C170 reference address and instruction buffer rank 2. Testing of the ICP rank 50 P input will not be done in IFT3/P. Testing of this input requires logic that is untested at this time. These inputs are tested in the order deemed appropriate for best isolation capability. Refer to the subsection for more information.

Section 6, Subsection 0

This subsection tests the BRANCH ADDRESS ADDER A input from the instruction buffer rank 2 (IB02). This input is enabled during an RNI sequence or during a C180 relative branch. This input will be tested using a C180 relative branch. The target address will be the same as the instruction address. This will place zeros on the B input to the adder. The instruction is structured such that the second execution will cause the current P value to be placed in RF4 and the Processor to halt. Various values of P will be used to verify the input path. The P values will be limited to physically addressable memory.

Hardware Tested:

- Instr Bfr RK2 input to Branch Address Adder.
- C180 Q times 2 input to Branch Address Adder.
- C180 Relative Branch control.

CM Data Used:

0 ----- 0

P Register Data Used (Hexadecimal):

0000 0000
FFFF FFFF
AAAA AAAA
5555 5555
8C8C 8C8C
7373 7373

Micrands Used:

Number	Description
71	If DAI condition D, Write RF7 with C170K from ICP, Write RF4 with P-current from ICP, Go to halt. Else, Set DAI condition D, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU-micrand, Write RF3 with REV-DATE from FU-micrand, Exit.
7A	Go to CSA 71.

Initialization:

- Write C180 decode RAMs with the relative branch bit set throughout.
- Write CSA 7A with a micrand branch to CSA 71.

Conditions:

- C.0
- a. Write the P register data to RF0.
 - b. Initialize five words of CM starting at test address.
 - c. Report the data written to RF0; informational.
- C.1
- a. Clear PFS registers and CM error logs.
 - b. Execute micrand 75 (write P register, clear DAI cond, exit).
 - c. Wait Processor halted (timed).
 - d. Read and report contents of RF4; error if not equal to condition 0 report.
- C.2
- a. Read PFS 86, 87, 88.
 - b. Report test status word; error if nonzero.
- C.3
- a. Read and report the contents of the micrand address register; informational.

60000284 A

IFT3/P - 41

Conditions:

- C.4
- a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit Subsection if last operand; else, repeat for next P register value.

Section 6, Subsection 1

This subsection tests the C180 Q times 2 input to the Branch Address Adder (BAA) B input Mux. This leg of the Mux is enabled when the instruction is a C180 relative Branch. To test this, a branch is executed with various values of Q. This subsection will also test the BAA input sign extension.

Micrands Used:

Number	Description
51	Initialize IF, Go to halt.
71	If DAI condition D, Write RF7 with C170K from ICP, Write RF4 with P-current from ICP, Go to halt. Else, Set DAI condition D, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7A	Go to CSA71.

Data Used (Hexadecimal):

0000
FFFF
AAAA
5555
8C8C
7373

Hardware Tested:

- C180 Q instruction portion input to BRANCH ADDRESS ADDER.
- BRANCH ADDRESS ADDER input sign extension.
- BRANCH ADDRESS ADDER Mux controls.

60000284 A

IFT3/P - 42

Initialization:

- Write RF00 and RF04 with zero.

Conditions:

- C.0
 - a. Write test instruction to CM Address 0.
 - b. Write 8 words of zero to CM starting at the branch target address.
 - c. Report Q value used; informational.
- C.1
 - a. Execute micrand 051 (Initialize IF).
 - b. Clear PFS registers and CM error logs.
 - c. Read PFS6, PFS7, PFS8.
 - d. Report test status word; error if nonzero.
- C.2
 - a. Execute micrand 75 (write IF P, clear DAI conditions, exit).
 - b. Wait Processor halted (timed).
 - c. Report contents of RF4; error if not equal two times the report of condition 0; bit 16 of Q should be extended.
- C.3
 - a. Read PFS registers 86, 87, and 89.
 - b. Report test status word; error if nonzero.
- C.4
 - a. Read and report contents of the micrand address register; informational.
- C.5
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Increment operand index.

Exit subsection if end of operands; else, repeat subsection for next operand.

Section 6, Subsection 2

This subsection tests the C170 Reference Address input to the Branch Address Adder A input Mux. This leg of the Mux is enabled with a C170 Relative branch. This subsection will set up the C170 decode RAM so that all C170 instructions are interpreted as relative branches. Control Store locations 100 through 3FF will be set up with the same micrand so that all op codes will be acted on the same. Data zero will be used as the "instruction" input to IF. This will keep the K times 4 input to the Adder a constant zero. The test will vary the value of the Reference Address and monitor the associated parity.

Hardware Tested:

- C170 RAC bits input to Branch Address Adder Mux.
- C170 RAM branch relative bit.

Micrands Used:

Number	Description
5C	Initialize IF and clear stream, Write RAC from RF1, Write RA+FL from RF2, Go to halt.
71	If DAI condition D, Write RF7 with C170K from ICP, Write RF4 with P-current from ICP, Go to halt. Else, set DAI condition D, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7A	Go to CSA71.

Reference Address Values Used (Hexadecimal):

(These values are truncated to obtain an address that is physically available.)

0000 0000
5555 5555
BABA BABA
FFFF FFFF

Initialization:

- Set RF02 register to FFFFFFFF(HEX).
- Write C170 decode RAMs with branch relative bit and 2 parcel size bit set throughout.
- Set C170 mode.

Conditions:

- C.0
 - a. Write Reference Address (RAC) from RF1 and RAFC from RF2.
 - b. Initialize CM starting at the branch target address.
 - c. Execute micrand 5C (write RAC from RF1, initialize IF).
 - d. Report value written to RAC; informational.
- C.1
 - a. Wait Processor halted.
 - b. Clear PFS Registers and CM error logs.
 - c. Read PFS registers 86, 87, 88.
 - d. Report IF status word; error if not equal zero.

Conditions:

- C.2
 - a. Execute micrand 75 (write RF0 to P, clear DAI conditions, exit).
 - b. Wait Processor halted (timed).
 - c. Read and report contents of RF4; error if not equal to condition 0 report; bits 32-39 will be cleared in the hardware.
- C.3
 - a. Read PFS registers 86, 87, and 88.
 - b. Report IF status word; error if nonzero.
- C.4
 - a. Read and report contents of RF3; informational, micrand address referenced by instruction.
- C.5
 - a. Read and report C170K portion of "instruction" executed; error if nonzero.
- C.6
 - a. Read and report contents of micrand address register; informational.
- C.7
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Increment operand index.

Exit subsection if end of operands; else, repeat subsection for next operand.

Section 8, Subsection 3

This subsection tests the CY170 K times 4 input to the Branch Address Adder B input Mux. This leg of the Mux is enabled when executing a C170 branch relative instruction. This input will be added to C170 RAC bits (reference address) in the Branch Address Adder. The result will be sent to LM and the P register.

This subsection will hold C170 RAC bits equal to zero. The C170 decode RAMs and Control Store will be set up to interpret all op codes as relative branches.

The value of the K portion of the instruction will be varied while parity is monitored.

Hardware Tested:

- K times 4 input to Branch Address Adder Mux.

60000284 A

IFT3/P - 45

Micrands Used:

Number	Description
51	Initialize IF, Go to halt.
52	Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
71	If DAI condition D, Write RF7 with C170K from ICP, Write RF4 with P-current from ICP, Go to halt. Else, Set DAI condition D, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7A	Go to CSA 71.

Data Used (Hexadecimal, C170 Instruction):

Instruction	
0000 0000	
0003 FFFF	
0002 AAAA	Instruction will be written
0001 5555	in parcel 2 and 3.
0000 8C8C	
0003 7373	

Initialization:

- Set RF1 = 0.
- Set RF2 = FF FFFF(HEX)
- Execute CSA 7C (write RF1 to RAC, write RF2 to RA+FLC Rgtr).

Conditions:

- C.0
 - a. Execute micrand 051 (initialize IF).
 - b. Clear PFS registers and CM error logs.
 - c. Read PFS6, PFS7, PFS8.
 - d. Report IF status word; informational.

60000284 A

IFT3/P - 46

Conditions:

- C.1
 - a. Write test instruction to CM Address 0.
 - b. Write 8 words of CM at target address to clear CM parity errors.
 - c. Report value of K field; informational.
- C.2
 - a. Execute micrand 075 (Write IF P, clear DAI condition register, exit).
 - b. Wait Processor halted (timed).
 - c. Read PFS Status (TSWD).
 - d. Read RF4.
 - e. Report Contents; error if not equal Condition 1 report times eight.
- C.3
 - a. Report IF status word; error if nonzero.
- C.4
 - a. Read and report contents of RF3; informational, value of micrand address referenced by last "instruction".
- C.5
 - a. Read and report contents of micrand address register informational.
- C.6
 - a. Report PFS Register 87 bits 19-23 in bits 59-83; informational.

Increment operand index.

Exit subsection if last operand, else, repeat for next operand.

Section 8, Subsection 4

The purpose of this subsection is to test the parity checkers associated with the Branch Address Adder B input multiplexer. These parity checkers are to verify that there is no error at the B input to the Branch Address Adder. Errors here and not at the Instruction Assembly mux/register will isolate a failure to a relatively small area of logic.

This subsection uses test mode bit 15 to verify that these parity checkers can detect and properly report an error. Several data patterns will be run in C170 mode.

Hardware Tested:

- Branch Address Adder B input parity checkers.
- BRAA B input parity mux.

80000284 A

IFT3/P - 47

Micrands Used:

Number	Description
93	Initialize IF. Write RAC from RF01. Write RAC+FLC from RF02. Clear DAI condition register. Read/Clear the MCR to RF12. Write P-Fetch from RF0. Exit.
58	Write RF7 with C170 K from ICP. Write RF8 with C180 JKQ from ICP. Write RF4 with P-Current from ICP. Go to Halt.
7A	Go to micrand 58.

Data Used:

The data used is generated from the iteration count as follows:

Instruction Bit	Iteration Count
9	2**8
10	2**7
11	2**6
15	2**5
18	2**4
24	2**3
13	2**2 XOR Parity of 9-11, 15
23	2**1 XOR Bit 18
29	2**0 XOR Bit 24

Initialization:

- Write zero to RF0 and RF01.
- Write CSA 7A with micrand branch to CSA 58.
- Clear the Test Mode register.
- Set the Test Mode Enable bit in P3DEC register.
- Write RF02 = F---F(HEX).
- Start execution at CSA 93 (purge errors).

Conditions:

- C.0
 - a. Write operand to CM address 00.
 - b. Set C180 mode.
 - c. If operand 200(HEX), set PTM bit 15.
 - d. Report data written to CM00 (information only).

80000284 A

IFT3/P - 48

Conditions:

- C.1
 - a. Clear P3 errors.
 - b. Clear M3 errors.
 - c. Start micrand execution at CSA 93.
 - d. Wait P3 halted (abort after delay).
 - e. Get test status word.
 - f. Master clear the processor.
 - g. Report the test status word (error if not as expected).
- C.2
 - a. Read and report contents of RF6 (error if not as expected).
- C.3
 - a. Set C170 mode.
 - b. Convert operand data to 170 format.
 - c. Write operand to address 0.
 - d. Report data written to CM00 (information only).
- C.4
 - a. Clear P3 errors.
 - b. Clear M3 errors.
 - c. Start micrand execution at CSA 93.
 - d. Wait P3 halted (abort after delay).
 - e. Get test status word.
 - f. Master clear the processor.
 - g. Report test status word (error if not as expected).
- C.5
 - a. Read and report contents of RF7 (error if not as expected).
- C.6
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit Subsection if end of operands.

Else, repeat Subsection for next operand.

SECTION 7 - TEST BRANCH ADDRESS ADDER

This section tests the BRANCH ADDRESS ADDER. Refer to the subsection documentation for further explanation.

Section 7, Subsection 0

This subsection tests the Branch Address Adder. This adder is used to form the target address for branch instructions. The desired inputs are selected with Mux logic ahead of the adder. The output of the adder is presented to LM for the next CM request and to the IF P register.

There are parity checkers on the output of the input multiplexers, at the adder inputs and at the output of the following register. These parity checkers will be monitored to detect any errors. The target P will also be read from ICP via microcode. This subsection will apply various operands to the A and B inputs of the Adder. The Adder output and parity checkers will be monitored to detect errors. The operands will be applied by executing a C180 Branch relative instruction.

Hardware Tested:

- Branch Address Adder.

Micrands Used:

Number	Description
71	If DAI condition D, Write RF7 with C170K from ICP, Write RF4 with P-current from ICP, Go to halt. Else, Set DAI condition D, Exit.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7A	Go to CSA 71.

IF Branch Address Adder:

	P (0 through 31) +	2Q (0 through 31)	Q (16 through 31)
1.	0000 AAAA	0000 AAAA	5555
2.	FFFF 5555	FFFF 5554	AAAA
3.	FFFF 5555	0000 AAAA	5555
4.	0000 AAAA	0000 5554	AAAA
5.	0000 AAAD	FFFF 5554	AAAA
6.	FFFF 5554	0000 AAAA	5555
7.	FFFF FFFF	0000 0000	0000
8.	0000 0001	FFFF FFFE	FFFF
9.	FFFE FFFF	0000 0000	0000
10.	FFFF 7F7F	0000 0000	0000
11.	FFFF BF8F	0000 0000	0000
12.	FFFF EFEF	0000 0000	0000
13.	FFFF 7F7F	0000 0000	0000
14.	FFFF FBFB	0000 0000	0000
15.	FFFF FEFE	0000 0000	0000
16.	0000 7F7F	FFFF 0102	8081
17.	FFFF 7F7F	0000 0202	0101
18.	FFFF 7F7F	0000 0404	0202
19.	FFFF 7F7F	0000 0808	0404
20.	FFFF 7F7F	0000 1010	0808
21.	FFFF 7F7F	0000 2020	1010
22.	FFFF 7F7F	0000 4040	2020
23.	FFFF FEFE	0000 8080	4040
24.	FFFF FEFE	0000 0404	0202
25.	FFFF FEFE	0000 0808	0404
26.	FFFF FBFB	0000 0A0A	0505
27.	FFFF FBFB	0000 1212	0909
28.	FFFF FEFE	0000 1010	0808
29.	FFFF EFEF	0000 2020	1010
30.	FFFF EFEF	0000 4040	2020
31.	FFFF DFDF	0000 4040	2020
32.	FFFF BF8F	0000 8080	4040
33.	FFFF DFDF	0000 8080	4040
34.	FFFF FFFF	FFFF FFFE	FFFF
35.	0000 0000	0000 0000	0000
36.	FFFF FFFF	0000 0000	0000
37.	FFFF FEFE	0000 0000	0000
38.	FFFF 4B4B	0000 0000	0000
39.	FFFF 2525	0000 0000	0000
40.	FFFF 9595	0000 0000	0000
41.	0001 FFFF	FFFF 5554	AAAA
42.	FFFF FFFF	0000 AAAA	5555
43.	0000 FFFF	FFFF 888A	DDDD
44.	FFFF FFFF	0000 EEEE	7777
45.	0000 FFFF	FFFF EFEE	F7F7
46.	FFFF FFFF	0000 FEFE	7F7F

+ The lower two bits of P are forced to zero.

Initialization:

- Write CSA 7A with a micrand branch to CSA 71.
- Write C180 decode RAM with the branch relative bit set throughout.

Conditions:

- C.0 a. Report operand number; informational.
- C.1 a. Write CM address to RF0.
b. Write data to CM address specified here.
c. Report CM address used; informational.
- C.2 a. Initialize CM at branch target address.
b. Report contents of CM address; informational.
- C.3 a. Execute micrand 075 (clear DAI condition bits, write RF0 to P, exit).
b. Wait Processor halted (timed).
c. Read and report contents of RF4; error if not equal the target address which is displayed in expected buffer.
- C.4 a. Read PFS 86, 87, 88.
b. Report test status word; error if nonzero.
- C.5 a. Read and report contents of RF3 (micrand address referenced by last instruction; informational.
- C.6 a. Read and report contents of micrand address register; informational.
- C.7 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if last operand, else, repeat subsection for next operand.

SECTION 8 - TEST ADDRESS OUT OF RANGE (AOR)

This section tests the C170 ADDRESS-OUT-OF-RANGE detection and report logic. This logic makes a comparison between the output of the BRANCH ADDRESS ADDER and the RACFLC register (RAC+FLC). If the add result is equal or greater than RAC plus FLC an address out of range (AOR) error should be reported.

SS00 Test the C170 ADDRESS-OUT-OF-RANGE detector.

SS01 Test AOR by sweeping through all addresses up to the maximum RA+FLC.

Section 8, Subsection 0

This subsection tests the C170 ADDRESS-OUT-OF-RANGE detector. This detector compares the RAC + FLC with the output of the Branch Address Adder. If the add result is equal or greater than RAC plus FLC an Address Out of Range (AOR) error should be reported. An Address Out of Range will cause an exchange interrupt to be initiated by Instruction Complete Control. For this subsection, Control Store will be set up so that the first instruction will "look" like a branch indexed. The microcode will load P-fetch from RF1 (target address) and exit waiting for the next instruction. The second instruction will load the current P register value from the pipe and halt. The data patterns used will cause an address-out-of-Range indication.

Operands Used to Test Address Out of Range:

R = RA+FLC

P = P register (target word address)

The test operands consist of a target word address and a reference (RAC) plus a field length (FLC). The test operands are generated in 4-bit groups starting with the least significant bits.

The operands are generated by setting all groups of less significance than the test group to R = F(HEX) and P = 0. Groups of greater significance than the test group will be set to zero. The test group will be set to one of the operands shown. Each operand set will cause an Address Out of Range.

Test operand: 0000 0000 0000 0000 0000 0000 (24-bit word address)
Test groups (a) (b) (c) (d) (e) (f)

NOTE

There are 4 bits per test group.

4-Bit Operand Group

R (RAC + FLC)	P (Target Address Hexadecimal)
0000 (0)	0000 (0)
0000 (0)	0001 (1)
0001 (1)	0010 (2)
0010 (2)	0011 (3)
0011 (3)	0100 (4)
0100 (4)	0101 (5)
0101 (5)	0110 (6)
0110 (6)	0111 (7)
0111 (7)	1000 (8)
1000 (8)	1001 (9)
1001 (9)	1010 (A)
1010 (A)	1011 (B)
1011 (B)	1100 (C)
1100 (C)	1101 (D)
1101 (D)	1110 (E)
1110 (E)	1111 (F)
1111 (F)	1111 (F)

Example:	R	P
	000AFF	000800
	002FFF	003000

NOTE

The target word address is equal or one greater than RA+FLC.

Micrands Used:

Number	Description
85	If DAI condition D, Write RF7 with C170 K from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-current from ICP, Go to halt. Else, set DAI condition D, Write P-fetch from RF13, Exit.
5C	Initialize IF and clear stream, Write RAC from RF1, Write RA+FL from RF2, Go to halt.

Micrands Used:

Number	Description
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU-micrand, Write RF3 with REV-DATE from FU-micrand, Exit.
7A	Go to CSA 85.

Initialization:

- Write CSA 7A with a micrand branch to CSA 85.
- Write zero to RF0 and RF1.
- Write C170 decode RAMs with the branch indexed bit set throughout.
- Set C170 mode.
- Write zero to CM address 0.

Conditions:

- C.0 a. Write the RA+FLC to be used in RF2.
 b. Report the value RA+FLC; informational.
- C.1 a. Write the target address to RF13.
 b. Report the target address; informational.
- C.2 a. Execute micrand 05C (write RA+FLC).
 b. Clear Processor errors.
 c. Execute micrand 075 (clear DAI condition bits, write RF0 to P,
 exit).
 d. Wait Processor halted (timed).
 e. Read PFS registers 86, 87, 88.
 f. Report test status word; error if not Processor halted or other
 bits nonzero.
- C.3 a. Read and report Micrand Address Register; error if not equal to
 1803(HEX).
- C.4 a. Read and report contents of RF3; error if not equal 380(HEX)
 micrand address referenced by the last instruction.
- C.5 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of test operands, else, repeat for next operand.

Section 8, Subsection 1

This subsection tests the C170 ADDRESS-OUT-OF-RANGE detector. This detector compares the RA+FLC with the output of the branch address adder. If the add result is equal or greater than RAC plus FLC, an address out of range (AOR) error should be reported.

This subsection tests address-out-of-range by referencing all addresses up to a preset RA+FL value. This is done by allowing all op codes to be interpreted as a do nothing instruction. When an address-out-of-range is detected, the current value of P will be written to RF4 and the processor will be halted. The P value read should equal the RA+FLC value.

NOTE

Improper reports of Address Spec error, Page Table Search without Find, First level instruction decode RAM parity errors, or Address-out-of Range could cause this subsection to fail. (Refer to EXECUTE OK signal to Control Store from Instruction Fetch.)

RA+FL Value Used (Hexadecimal):

1	70F	0F0F
2	0F0	F0F0

Micrands Used:

Number	Description
10	Exit-unconditionally if IB12 valid, load RF3 = CSA entry. (If not AOR, go to 7A. If AOR, go to 55.)
7A	Go to 10.
55	Write RF4 from P-current, Halt.

Initialization:

- Write zero to RF0 and RF1.
- Write C170 decode RAMs with Size = 30 bits.
- Write CSA 7A with a micrand branch to CSA 10.
- Write CSA 380 with a micrand to write RF3 = 380(HEX) and branch to CSA 55.

Conditions:

- C.0 a. Write the RA+FLC value to RF2.
 b. Report RA+FLC value; informational.
- C.1 a. Write RAC and RA+FLC (micrand 5C).
 b. Clear Central Memory error logs.
 c. Clear Processor errors.
 d. Execute micrand sequence 52 (write P, and exit).
 e. Wait Processor halted.
 f. Get and report test status word; error if nonzero other than instruction assembly register parity errors.
- C.2 a. Read and report the contents of micrand address register; error if not 1803(HEX).
- C.3 a. Read and report RF3 contents; error if not equal 380(HEX).
- C.4 a. Read and report contents of RF4; error if not equal to the RA+FLC value.
- C.5 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands, else, repeat subsection for next operand.

SECTION 9 - MISCELLANEOUS TESTS

This section consists of tests of a miscellaneous nature. Some of these tests have been held until this section because of isolation considerations. Refer to the appropriate subsection description for additional explanation.

Section 9, Subsection 0

This subsection tests the gating of the PARCEL 3 SAVE REGISTER and the enabling of errors detected there. This register holds Parcel 3 of the CM word currently being parcelled when a request has been made and accepted for the next CM word. Parity errors detected in this register are not enabled until LM has accepted the second CM request following a Master Clear signal. At this time, the information in the register should be valid.

This subsection tests the parcel 3 save register parity detection and reporting. This logic will be tested for three conditions: a parity report from the instruction assembly register parcel 3 and not the parcel 3 save register, a parity error report from both the instruction assembly register and the parcel 3 save registers, and that the parity error report from parcel 3 save is seen only once.

The first condition will be tested by setting the IF P-register and halting Processor. IF will prefetch instructions until the IF instruction buffers are full. Parcel 3 of the second CM word referenced will have a byte parity error. The result will be a parity error in instruction assembly register parcel 3, but not in the parcel 3 save register.

The second condition is tested by executing five instructions (5 exits) starting at the same CM address as the previous condition. This should cause a parity error in both the instruction assembly register and the Parcel 3 save registers.

The third condition is tested by repeating condition 2, clearing Processor errors, and then getting status. There should be no instruction assembly register or Parcel 3 save parity errors.

Hardware Tested:

- PARCEL 3 SAVE REGISTER parity check.
- Gating of parity on PARCEL 3 REGISTER.

Micrands Used:

Number	Description
51	Initialize IF, Go to halt.
52	Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.

Micrands Used:

Number	Description
53	Initialize IF and clear stream, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Go to halt.
60	If DAI condition A, Go to 60-1. Else, set DAI condition A, Write RF5 from RF3, Write RF6 with C180 JKQ from ICP, Exit.
60-1	If DAI condition B, Go To 60-2. Else, Set DAI conditions A and B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.
60-2	If DAI condition C, Go To 60-3. Else, Set DAI conditions A, B and C, Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
60-3	If DAI condition D, Go To 60-4. Else, Set DAI conditions A, B, C and D, Write RFB from RF3, Write RFC with C180 JKQ from ICP, Exit.
60-4	Write RF4 with P-Current from ICP, Go To Halt.
80	Write DAI condition register from RF10, Go to halt.
7A	Go to CSA 80.

Data Used (Hexadecimal):

CM	00	00	----	00	
	01	00	----	01	(Byte 7 parity check)
	02	0	----	0	
	03	0	----	0	
	04	0	----	0	
	05	0	----	0	
	06	0	----	0100	(Byte 8 parity check)
	07	0	----	0	
	08	0	----	0	
	09	0	----	00	
CM	0A	0	----	00	

Initialization:

- Set Central Memory into SECEDED mode.
- Set Write Check bits (Central Memory).
- Write data to CM (see data used).
- Set Central Memory parity mode (disable SECEDED); clear Write check bits.
- Write zeros throughout the C180 decode RAM.
- Write CSA 7A with a micrand branch to CSA 80.

Conditions:

- | | |
|-----|---|
| C.0 | a. Execute Micrand 51 (initialize Instruction Fetch). |
| | b. Master clear Processor. |
| | c. Write starting P value to RF0. |
| | d. Clear Processor errors. |
| | e. Execute Micrand 53 (write RF0 to P, Halt). |
| | f. Wait Processor halted. |
| | g. Get test status word. |
| | h. Report test status word; error if not proper parcel 3 parity error (byte 6 or 7). |
| C.1 | a. Master clear Processor. |
| | b. Set DAI conditions A, B, C, and D (Micrand 80). |
| | c. Execute micrand 52 (write RF0 to P, exit). |
| | d. Wait Processor halted. |
| | e. Get test status word. |
| | f. Report test status word; error if parity error reported on condition 0 is not in Instruction Assembly register and parcel 3 save register. |
| C.2 | a. Master clear. |
| | b. Set DAI condition bits A, B, C, and D (Micrand 80). |
| | c. Clear Processor errors. |
| | d. Execute Micrand 52 (write RF0 to P, exit). |
| | e. Clear Processor errors. |
| | f. Get test status word. |
| | g. Report test status word; error if nonzero. |
| C.3 | a. Read and report contents of RF4; error if not equal the value of last instruction executed. |
| C.4 | a. Read and report the contents of the micrand address register; informational. |
| C.5 | a. Report PFS Register 87 bits 19-23 in bits 59-63; informational. |

Exit subsection if end of subsection, else, repeat subsection for next operand.

Post Subsection Operations:

- Return Central Memory to SECDED mode.

Section 9, Subsection 1

This subsection will verify that the parity checkers following Rank 2 and 12 of the INSTRUCTION BUFFER can report parity errors.

The parity checker following Rank 2 checks bits 3, 12 through 40 of the instruction assembly buffer. Any errors will be reported via PFS register 87 bits 8 through 11 (status word bits 32 through 35).

The parity checkers following Rank 12 check bits 0 through 22 and 30 through 47 of both IB12 paks. This represents checks on P register bits 32 through 63 and instruction assembly buffer bits 3 through 40. Any errors are reported via PFS register 86 bits 43 through 47 and bits 51 through 54.

The appropriate parity bits are forced to zero by PTM bit 15. The data will be varied to prove the parity checkers.

Micrands Used:

Number	Description
51	Initialize IF, Go to halt.
5C	Initialize IF and clear stream, Write RAC from RF1, Write RA+FLC from RF2, Go to halt.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
85	If DAI condition D, Write RF7 with C170 K from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-current from ICP, Go to halt. Else, set DAI condition register, Write P-fetch from RF13, Exit.
7A	Go to CSA 85.

60000284 A

IFT3/P - 61

Data Used (Hexadecimal):

P register values/instruction values (Hexadecimal)

0	0000	0000	0000	0000
1	0000	0004	0000	0000
2	0000	0100	0000	0000
3	0001	0000	0000	0000
4	0000	0000	0000	0000
5	0000	0000	0000	0004
6	0000	0000	0000	0400
7	0000	0000	0004	0000
8	0000	0000	0400	0000

Initialization:

- Set C170 mode.
- Write C170 decode RAMs with branch indexed and size equal to 30 bits.
- Set Processor test mode bit 15.
- Write CSA 7A with micrand branch to CSA 85.
- Write RAC = 0, RA+FLC = FFFFFFFF(HEX) (Micrand 5C).

Conditions:

- C.0
 - a. Execute micrand 51 (initialize IF).
 - b. Clear PFS registers.
 - c. Write P register address to RF0.
 - d. Write the appropriate CM data to the address contained in RF0. Write zero to that address +1.
 - e. Report operand number; informational.
- C.1
 - a. Execute micrand 75 (write RF0 to P Register, clear DAI conditions, exit).
 - b. Wait Processor halted (timed).
 - c. Read PFS registers.
 - d. Report test status word (if not equal to expected value).
- C.2
 - a. Read and report contents of RF4; error if not equal P value appropriate to operand number reported in C.0.
- C.3
 - a. Read and report contents of RF7; error if not equal to C170K portion of instruction appropriate to operand number reported in C.0.
- C.4
 - a. Read and report contents of RF3; informational contains last micrand address referenced by virtual instruction.
- C.5
 - a. Report test data used; informational.
- C.6
 - a. Read and report contents of micrand address register; informational.

60000284 A

IFT3/P - 62

Conditions:

C.7 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands, else, repeat subsection for next operand.

Note that the PTM register will be cleared prior to leaving the subsection.

Section 9, Subsection 2

This subsection tests the BDP input to the INSTRUCTION MUX (IAHY). This Mux will select one of four inputs for IAHY bits 1, 3 through 11, 41 through 43, 49 and 50. It will select one of three inputs for IAHY bits 12 through 40. The BDP leg of this Mux is selected for BDP descriptors only (i.e., only after a BDP function has been recognized).

This subsection will vary the bits used to generate the micrand address for the descriptor(s). The referenced micrand will store its address to the register file and branch to a micrand which will load the remaining bits to another location of the register file.

Hardware Tested:

- BDP leg of INSTRUCTION MUX
- Saved BDP OPCODE register
- BDP Control

Micrands Used:

Number	Description
64	If DAI condition B, Go to 64-1. Else, Set DAI conditions A and B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.
64-1	If DAI Condition C, Go To 64-2. Else, Set DAI Condition A, B and C Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
64-2	If DAI Condition D, Go To 64-3. Else, Set DAI Conditions A, B, C and D, Write RFB from RF3, Write RFC with C180 JKQ from ICP, Exit

60000284 A

IFT3/P - 63

Micrands Used:

Number	Description
64-3	Write RF4 with P-Current from ICP, Go To Halt.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU-micrand, Write RF3 with REV-DATE from FU-micrand, Exit.
7A	Go to CSA 64.

Data Used is hexadecimal:

Number	BDP INSTR	DESCR1	Micrand Address of DESCR1
1	00FF FFFF	0000 0000	200
2	0-----0	80FF FFFF	220
3	AAFF FFFF	80A-----A	2AA
4	55F-----F	0055 5555	245
5	00-----0	5-----5	200
6	40-----0	0-----0	200
7	00-----0	80-----0	220
8	20-----0	0-----0	200
9	100-----0	0-----0	240
10	080-----0	0-----0	208
11	040-----0	0-----0	204
12	020-----0	0-----0	202
13	010-----0	0-----0	201

Initialization:

- Write CSA 7A with micrand branch to CSA 64.
- Write ones throughout the C170 decode RAMs.
- Write C180 decode RAM with the BDP start bit set throughout.

Conditions:

- C.0
- Write BDP pseudo instruction to CM address 0.
 - Clear PFS registers.
 - Execute micrand 75 (Write P register, clear DAI conditions, exit).
 - Wait Processor halted (timed).
 - Read PFS registers 88, 87, 88.
 - Report test status word; error if nonzero.
- C.1
- Read and report contents of RF7; error if not equal to the micrand address generated by the BDP instruction; the expected value is in the expected buffer.

60000284 A

IFT3/P - 64

Conditions:

- C.2 a. Read and report the contents of RF9; error if not equal to the micrand address generated by the first BDP descriptor; the expected value is displayed from the expected buffer.
- C.3 a. Read and report contents of the RFA; error if not equal to bits 8 through 31 of the first descriptor, sign extended; the expected value is displayed from the expected buffer.
- C.4 a. Report the operand number; informational.
- C.5 a. Read and report the contents of the micrand address register; informational.
- C.6 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands; else, repeat for next operand.

Section 9, Subsection 3

This subsection tests the BDP control logic using two descriptors. Instruction Fetch logic "detects" a BDP instruction by bit 4 of the C180 decode RAM. Once this "detection" is made, descriptors (32 bits long) are parcelled out based on the condition of bit 3 (two descriptors) of the C180 decode RAM. The presence of C180 RAM bit 3 without C180 RAM bit 4 (BDP Start) has no detectable effect on parcelling of instructions.

The BDP control will be tested using one and two descriptors. Instruction bits 1 and 2 will be varied and the effect on IF parity will be verified.

Hardware Tested:

- BDP control.

Micrands Used:

Number	Description
64	If DAI condition B, Go to 64 through 1. Else, Set DAI conditions A and B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.

Micrands Used:

Number	Description
64-1	If DAI Condition C, Go To 64-2. Else, Set DAI Condition A, B and C Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
64-2	If DAI Condition D, Go To 64-3. Else, Set DAI Conditions A, B, C and D, Write RFB from RF3, Write RFC with C180 JKQ from ICP, Exit
64-3	Write RF4 with P-Current from ICP, Go To Halt.
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7A	Go to CSA 64.

Operands Used:

0	9600,0000,3333,3333,4444,4444,FFFF,FFFF
1	9600,0000,5555,5555,6666,6666,FFFF,FFFF
2	9600,0000,7777,7777,8888,8888,FFFF,FFFF
3	9600,0000,9999,9999,AAAA,AAAA,FFFF,FFFF
4	9500,0000,3888,8888,80CC,CCCC,FFFF,FFFF
5	9500,0000,DDDD,DDDD,EEEE,EEEE,FFFF,FFFF
6	96FF,FFFF,0000,0000,1111,1111,FFFF,FFFF
7	97EE,EEEE,2222,2222,3333,3333,FFFF,FFFF
8	98DD,DDDD,4444,4444,5555,5555,FFFF,FFFF
9	9900,0000,6666,6666,7777,7777,FFFF,FFFF
A	97BB,8888,0888,8888,9999,9999,FFFF,FFFF
B	98AA,AAAA,2AAA,AAAA,8888,8888,FFFF,FFFF
C	9999,9999,4CCC,CCCC,DDDD,DDDD,FFFF,FFFF
D	9688,8888,6EEE,EEEE,7FFF,FFFF,FFFF,FFFF
E	9A77,7777,0123,4567,7654,3210,FFFF,FFFF
F	9B66,6666,5555,5555,2AAA,AAAA,FFFF,FFFF
10	9C55,5555,3333,3333,6666,6666,FFFF,FFFF
11	9644,4444,CCCC,CCCC,3333,3333,FFFF,FFFF
12	9A33,3333,8888,8888,7777,7777,FFFF,FFFF
13	9B22,2222,EEEE,1111,2222,CCCC,FFFF,FFFF
14	9C11,1111,89AB,CDEF,FEDC,BA98,FFFF,FFFF

C180 Decode RAM:

This RAM contains zeros with exception of the following addresses:

95, CB, D8, EB, F8	-	C180 BDP (R80 (4)) set
CA, D7, EA, F7	-	C180 Two Descr BDP set
96-9C, CC, D9, EC, F9	-	C180 Two Descr BDP, C180 Two Descr Bit BDP

Initialization:

- Write C180 decode RAMs.

Conditions:

- C.0
 - a. Write operand to CM address 0.
 - b. Clear PFS registers.
 - c. Execute micrand 052 (Write RF0 to IF P Register, clear DAI condition bits, exit).
 - d. Wait Processor halted (timed).
 - e. Read PFS registers 86, 87 and 88.
 - f. Report test status word; error if nonzero.
- C.1
 - a. Read and report contents of RF7; error if not equal the expected micrand address generated by the first instruction.
- C.2
 - a. Read and report contents of RF8; error if not equal bits 8 through 31 of first instruction executed.
- C.3
 - a. Read and report contents of RF9; error if not equal the expected micrand address formed by the first descriptor.
- C.4
 - a. Read and report the contents of RFA; error if not equal bits 8 through 31 of the first descriptor.
- C.5
 - a. Read and report the contents of RF3; error if not equal to the micrand address expected to be formed by the second descriptor.
- C.6
 - a. Read and report the contents of RFC; error if not equal to bits 8 through 31 of the second descriptor.
- C.7
 - a. Read and report contents of RF4; error if not equal to the address of the instruction following the BDP instructions.
- C.8
 - a. Report operand number; informational.
- C.9
 - a. Read and report contents of micrand address register; informational.
- C.10
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

60000284 A

IFT3/P - 67

Conditions:

Exit subsection if end of operands, else, repeat subsection for next operand.

Section 9, Subsection 4

This subsection tests the PARCEL REQUEST LOGIC. There are three conditions that will generate a parcel request. A parcel request is generated: during parcel 1 if the instruction is a two parcel instruction and not a branch, during parcel 2 if the instruction is not a branch, during parcel 3 if parcel 3 was the target address of a branch instruction.

In this subsection, an instruction sequence testing each of these conditions is stored in CM address 0. In CM address 01 is stored a word with parity errors. If this word is referenced, parity errors should be detected at the CM and at the INSTRUCTION ASSEMBLY REGISTER of IF.

The test consists of five passes of the subsection with the different instruction sequences in CM address 0. The five sets of sequences establish the following conditions: branch in parcel 2 with single parcel instructions preceding, branch in parcel 1, branch in parcel 3 with a 2 parcel instruction in parcel 1, branch in parcel 3 with single parcel instructions in parcels 0 through 2, branch to parcel 3 originating in parcel 1. The first two sequences should produce no reference to CM address 01. The remaining sequences should all produce references to CM address 01 and produce errors as a result.

Hardware Tested:

- PARCEL REQUEST logic.
- PARCEL DECODE logic.

Micrands Used:

Number	Description
75	Clear DAI condition register, Initialize IF, Write P-fetch from RF0, Write RF4 with REV-DATE from FU micrand, Write RF3 with REV-DATE from FU micrand, Exit.
7A	Go to CSA 92.
91	Write RF4 with C180 JKQ from ICP, Write P-fetch from RF4, Go to 92.

60000284 A

IFT3/P - 68

Micrands Used:

Number	Description
92	Write RF3 with P-current from ICP, IF DAI condition A, Go to 92-1. Else, set DAI condition A, Write RF5 from RF3, Write RF8 with C180 JKQ from ICP, Exit.
92-1	IF DAI condition B, Go to 92-2. Else, set DAI conditions A and B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.
92-2	IF DAI condition C, Go to 92-3. Else, set DAI conditions A, B and C, Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
92-3	IF DAI condition D, Go to 92-4. Else, set DAI conditions A, B, C and D, Write RFB from RF3, Write RFC with C180 JKQ from ICP, Exit.
92-4	Write RF4 with P-current from ICP, Go to halt.

CM Data Used (Hexadecimal):

Operand Number	CM Addr	Operand
0	00 01	7100 7300 F400 0000 0000 0000 0000 0101
1	00 01	7100 F400 0000 7100 0000 0000 0000 0101
2	00 01	7100 8300 F400 0000 0000 0000 0000 0101
3	00 01	7100 7300 7300 F400 0000 0000 0000 0101
4	00 01	7100 F400 0008 7100 0000 0000 0000 0101

80000284 A

IFT3/P - 69

Initialization:

- Write zeros to CM 02 through 10(HEX).
- Write C180 decode RAM address 81 with the Branch indexed bit.
- Write zero to RF0.
- Write CSA 7A with a micrand branch to CSA 92.
- Write CSA 181 with a micrand branch to CSA 91.

Conditions:

- C.0 a. Report operand number; informational.
- C.1 a. Clear PFS register.
b. Clear Central Memory error logs.
c. Write instruction sequence to CM 00.
d. Execute micrand 75 (Write RF0 to IF P, clear DAI condition bits, exit).
e. Wait Processor halted (timed).
f. Read PFS registers 86, 87, 88.
g. Report test status word; error if nonzero for operands 0 and 1; error if no Instruction Assembly register parity errors for operands 2 through 4.
- C.2 a. Read and report contents of Central Memory uncorrected error log 1; error if nonzero for operands 0 and 1; error if zero for operands 2 through 4.
- C.3. a. Read and report contents of Central Memory uncorrected error log 2; informational.
- C.4 a. Read contents of RF5.
b. Report contents of RF5; error if nonzero, address of first instruction executed.
- C.5 a. Read contents of RF7.
b. Report contents of RF7; error if not equal address of expected second instruction.

Operand Number	RF7
0-3	2
4	6
5	0

- C.6 a. Read and report contents of RF9; error if not equal the expected address of the third instruction.

Operand Number	RF9
0 or 3	4
1 or 4	0
2 or 5	6

80000284 A

IFT3/P - 70

Conditions:

- C.7 a. Read and report contents of RFB; error if not equal to the expected address of the third instruction executed.

Operand Number	RFB
0	0
1	2
2	0
3	6
4	6
5	0

- C.8 a. Read and report contents of micrand address register; informational.

- C.9 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands; else repeat subsection for next operand.

Section 9, Subsection 5

This subsection tests the INSTRUCTION BUFFER ranks 3, 10, and 11. It will also verify the buffer valid bits for those ranks.

Buffer ranks 2, 3, 10, 11, and 12 provide buffering for five instructions in IF. Instruction Buffer rank 12 feeds the Instruction Control Pipe (ICP) and Control Store (CS). If all buffer ranks are empty when the Instruction Assembly register becomes valid, buffer rank 12 will be loaded with rank 2 and the transfer from rank 2 to rank 3 will be blocked. Thus the first instruction buffered will bypass the earlier ranks.

The test consists of a sequence of instructions which will store predefined information in register file locations. The instruction addresses will be varied to provide a test of bits 0 through 17 of the buffers. This will also provide a test of the IB12 Mux.

Hardware Tested:

1. INSTRUCTION BUFFERS 02, 03, 10, and 11.
2. Buffer valid control (including fanouts).
3. IB12 Mux.
4. Data to rank 2, 12 enables.

Data Used (Hexadecimal):

Operand Number	CM Word Adrs	Data
0	0----0	7FFF 7FFF 0000 5555
1	1FFFF	7FFF 7FFF 0000 5555
2	0AAAA	7FFF 7FFF 0000 5555
3	15555	7FFF 7FFF 0000 5555
4	0----0	0000 0000 FFFF 0000
5	0----0	7FFF 7FFF 5555 AAAA
6	0----0	7FFF 7FFF AAAA 3333
7	0----0	7FFF 7FFF BCBC 7373

Micrands Used:

Number	Description
2	If not halt flag, set halt flag and branch to CSA 002; else, go to CSA 003.
3	If halt flag, branch to CSA 003; else, exit.
54	Move JKQ portion of operand from ICP to RF8.
55	Move P from ICP to RF4 and branch to CSA 002.
53	Initialize IF, write RF0 to P-fetch, halt.
7A	Go to CSA 54.

Initialization:

- Write CSA 7A with micrand branch to CSA 54.
- Execute micrand 051 (Initialize IF and halt).
- Write zero throughout the C180 decode RAM.

Conditions:

- C.0
- a. Clear PFS registers.
 - b. Write test data to CM address X.
 - c. Write value X to RF0.
 - d. Report value X; informational.
- C.1
- a. Execute micrand 053 (Move RF0 to IF P, halt).
 - b. Set N = 0.
 - c. Start Processor.
 - d. Wait Processor halted.
 - e. Increment N.
 - f. If N less than 3, go to 3.
 - g. Read PFS registers 86, 87, 88.
 - h. Report test status word; error if nonzero.

Conditions:

- C.2 a. Read and report contents of RF4; error if not equal to X+4; see condition 0 report.
- C.3 a. Read and report contents of RF8; error if not equal JKQ portion of instruction.
- C.4 a. Read and report contents of RF3; error if not equal the expected micrand address; see the expected register display.
- C.5 a. Report operand number; informational.
- C.6 a. Read and report contents of micrand address register; informational.
- C.7 a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands; else, repeat subsection for next operand.

Section 9, Subsection 6

This subsection does some testing of the address spec error logic. Address spec errors in C170 mode with the exception of a two-parcel instruction in parcel 3 are blocked in the instruction assembly hybrid Mux.

The exception condition is not tested here. This subsection will execute C170 mode sequences of one and two parcel lengths to prove that the blocking signals do not force an error condition. The control store address generated by each sequence will be monitored.

Hardware Tested:

- Address specification error detection.
- Rank 12 parcel 3 ASE (IB020L bit 01, NRNK 12 bit 31).

Micrands Used:

Number	Description
75	Clear DAI condition bits. Initialize IF. Write P-fetch from RF0. Write RF4 from Rev-Date. Write RF3 from Rev-Date. Exit.
100-37F	Write RF3 with constant equal to CSA. Go to 7A.
7A	Go to 60.

60000284 A

IFT3/P - 73

Micrands Used:

Number	Description
60	If DAI condition A, go to 64. Else, set DAI condition A, Write RF5 from RF3. Write RF6 with C180 JKQ. Exit.
64	If DAI condition B, go to 64-1. Else, set DAI conditions A and B. Write RF7 from RF3. Write RF8 from C180 JKQ. Exit.
64-1	If DAI condition C, go to 64-2. Else, set DAI conditions A, B, and C. Write RF9 from RF3. Write RFA from C180 JKQ. Exit.
64-2	If DAI condition D, go to 64-3. Else, set DAI conditions A, B, C, and D. Write RFB from RF3. Write RFC from C180 JKQ. Exit.
64-3	Go to 55.
55	Write RF4 from P-current. Halt.

Initialization:

- Set RF0 equal zero.
- Set Size = 30 bits throughout C170 Odd RAM.
- Set zero throughout the C170 Even RAM.
- Write CSA 7A with a micrand branch to CSA 60.
- Write CSA 380 with a spin micrand.
- Set RF1 equal zero.
- Set RF2 equal zero.
- Execute Micrand 5C (write RAC from RF1, write RA+FLC from RF2).
- Set C170 mode.

Conditions:

- C.0 a. Write operand to CM address 0.
b. Report contents of CM 0; informational.
- C.1 a. Write operand to CM address 0.
b. Report contents of CM address 00; informational.

60000284 A

IFT3/P - 74

Conditions:

- C.2
 - a. Clear Processor errors.
 - b. Execute Micrand 75 (clear DAI conditions, write RF0 to P, exit).
 - c. Wait Processor halted.
 - d. Get test status word (GTS subroutine).
 - e. Read and report contents of RF4; error if not equal C(HEX).
- C.3
 - a. Report test status word; error if nonzero.
 - b. Report value written; informational.
- C.4
 - a. Read and report contents of RFA; error if nonzero.
- C.5
 - a. Read and report contents of RF3; error if not equal 300(HEX).
- C.6
 - a. Read and report contents of the micrand address register; error if not 1803(HEX).
- C.7
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Post Subsection Operations:

- Restore CSA 380 with micrand branch to 7A.
- Set C180 mode.

Section 9, Subsection 7

This subsection tests the address spec error detection in IF. An odd byte address will be sent to IF through OPI using an index branch. The odd byte address is expected to set address-spec-error in IF. This signal will cause the signal EXECOK to control store to drop. This should cause control store to branch to CSA 380.

Address spec error will also be generated using address bit 32.

Hardware Tested:

- Address specification error detection.
- Rank 12 PARCEL 3 ASE (IB020L bit 01, NRNK 12 bit 31).
- EXECOK signal.

Micrands Used:

Number	Description
75	Clear DAI condition bits.
52	Initialize IF, clear stream, LRW-P-Fetch from RF0. Write RF4 from micrand = Rev-Date. Exit.

60000284 A

IFT3/P - 75

Micrands Used:

Number	Description
85	If DAI-COND-D, go to 85-1. Else, set DAI-COND-D, LRW-P-Fetch from RF13, Exit.
85-1	Clock Condition Register. Write RF7 from instruction K170, Write RF6 from instruction JKQ. Write RF4 from P-current. Go to halt.
7A	Go to 85.

Initialization:

- Write RF0 with zero.
- Set Branch Indexed bit throughout the C180 decode RAM.
- Write CSA 7A with a micrand branch to CSA 85.
- Initialize five CM words starting at CM 00.

Conditions:

- C.0
 - a. Write branch target address to RF13.
 - b. Report address; informational.
- C.1
 - a. Execute Micrand 75 (clear DAI conditions, write RF0 to P, exit).
 - b. Wait Processor halted.
 - c. Report contents of Processor status summary register; informational.
- C.2
 - a. Read and report contents of RF3; error if not equal 380(HEX).
- C.3
 - a. Read and report the contents of the micrand address register; error if not equal 1803(HEX).
- C.4
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit subsection if end of operands, else, repeat subsection for next operand.

Section 9, Subsection 8

This subsection tests the IF-PURGE signal into instruction fetch. In the process, it uses the PURGE-IN-STACK flag, Control Store Address generation and the C170 first level decode STORE bit.

60000284 A

IFT3/P - 76

A pseudo instruction sequence is executed that will vary the conditions of:

- a. Instruction mux bit 49 (C170 store or C180 opcode bit 7),
- b. C170/180 mode, and
- c. Purge-in-stack

If the PURGE-IN-STACK bit is set and a C170 STORE instruction is detected, the instruction fetch should halt and bit 2**0 of the normal Control Store Address vector should be logged. These results will be verified.

Hardware Tested:

- IFPURG signal into parcel request logic in IF.

Micrands Used:

Number	Description
94	Write PURGE-IN-STACK flag from RF07. Write zero to RF3. Go to m.93.
93	Initialize -IF. Write RAC register from RF01. Write RAC+FLC from RF02. Clear the DAI condition bits. Clear the MCR. Write P-fetch from RF00. Exit. (Wait IB12 valid.) (When IB12 becomes valid, Control Store will begin execution at some CSA within the range 100(HEX) through 3FF(HEX). The microcode at these addresses will cause RF3 to equal the CSA executed and branch to CSA 7A.)
7A	Branch to CSA 8A.
8A	If DAI condition D is set, go to 8A-1; else, write RF04 from RF03. Set DAI condition D. Exit.
8A-1	Write RF09 from C170 K instruction portion. Halt.

Test Operands:

Operand Index	Purge-In-Stack	IAH4 Mux Bit (49)	Machine Mode
0	0	0	C170
1	0	1	C170
2	1	0	C170
3	1	1	C170
4	1	1	C180

Initialization:

- Write RF02 = (all set).
- Write RF00 = zero.
- Write RF01 = zero.
- Clear the C180 first level decoder RAM.
- Write 1501(HEX) to C170 first level decode RAM address 53(HEX).
- Write branch micrand to CSA 7A. (Branches to CSA 8A.)
- Write proper operand set to CM address 0 and 1.
- Write desired PURGE-IN-STACK flag to RF07.
- Set desired machine mode.

Conditions:

- C.0
 - a. Clear P3 errors.
 - b. Clear M3 errors.
 - c. Start P3 execution at CSA 94.
 - d. Read P3 Status Summary register.
 - e. Repeat 4, until P3 halts (continue after some delay).
 - f. Repeat P3 Status Summary register value. Error if P3 halted bit not as expected (set except for operand 3).

- C.1
 - a. Get test status word.
 - b. Report test status word (error if not zero).

- C.2
 - a. Read and report contents of RF04 (error if not as expected).

I	RF4	RF3
0	300	300
1	350	300
2	300	300
3	351	351
4	105	102

- C.3
 - a. Read and report contents of RF03 (error if not as expected).

- C.4
 - a. Read and report contents of RF09 (informational only).

- C.5
 - a. Report PFS Register 87 bits 19-23 in bits 59-63; informational.

Exit Subsystem if end of operands, else repeat for next operand.

GLOSSARY

A

Abs

Absolute.

AC or A/C

Address Control.

ACMIC

Address Control Functional Micrand.

ADATB/RDSB/B

Refers to B Operand data in Multiply/ Divide unit test MDT3/P.

ADATC/RDSC/C

Refers to C Operand data in Multiply/ Divide unit test MDT3/P.

Adder

Refers to Large Adder in ALN test ANT3/P. Refer to Multiply Final Adder in test MDT3/P.

Adder Bit 47 FF

Large Adder Result bit 47 flip-flop in ALN test ANT3/P.

Adrs

Address.

ALN

Arithmetic and Logical Network.

ALU

Arithmetic and Logical Unit.

60000284 A

IFT3/P - A1

AOR

Address-Out-of-Range.

Arith

Arithmetic.

ASCII

American Standard Code for Information Interchange.

ASE

Address Specification Error.

ASID

Active Segment Identifier.

Assy

Assembly.

Aux

Auxiliary.

B

Refers to the operand input to ALN on the RSDB data path. Used in ALN test ANT3/P.

BCD

Binary Coded Decimal.

BD

Branch Delta.

BDP

Business Data Processor.

60000284 A

IFT3/P - A2

Bfr

Buffer.

Bin

Binary.

Bkpt

Breakpoint.

1BN

Byte Number.

Bound

Boundary.

Br

Branch.

C

Refers to the operand input to the ALN on the RDSC data path. Used in ALN test ANT3/P.

CA

Condition Action.

CABR

Cache Address Buffer Register.

CAE

Condition Action Extension.

CAR

Cache Address Register.

60000284 A

IFT3/P - A3

CBP

Code Base Pointer.

CCR

Clock Condition Register.

CEL

Corrected Error Log.

C.GT.B

C Greater than B signal from the SKG in ALN test ANT3/P.

Chan

Channel.

C1k

Clock.

C1r

Clear.

CM

Central Memory.

CMC

Central Memory Control.

Cnt

Count.

Cntr

Counter.

60000284 A

IFT3/P - A4

Coef

Coefficient.

Com

Common.

Comp

Complement.

Cond

Condition.

Conf1

Conflict.

Cont

Control.

Conv

Convert, Converter.

Cor

Corrected.

CPU

Central Processing Unit.

ICS

Condition Sensing.

CSA

Control Store Address.

80000284 A

IFT3/P - A5

CST

Control Store.

CW1

Control Word 1.

CW2

Control Word 2.

DAI

Data Interchange.

DCDR

Decoder.

DCI

Data control information. Two bytes of information that give the maintenance register address.

DEC

Dependent Environment Control, Decimal.

Decr

Decrement.

Descr

Descriptor.

Destn

Destination.

Disassy

Disassembly.

80000284 A

IFT3/P - A6

Div

Divide.

Divisor Rgtr

Divide network register internal to LSI CI divide arrays.

Dlyd

Delayed.

DMR

Debug Mask Register.

Dsb1

Disable.

DSS

Data Subfunction Select Micrand Field.

EBCDIC

Extended Binary Coded Decimal Inter- change Code.

ECC

Error Correction Code.

ECR

C170 Exit Mode Condition Register.

EI

Environment Interface.

EMMR or EMR

Exit Mode Mask Register.

60000284 A

IFT3/P - A7

Enb1

Enable.

Encdr

Encoder.

ESE

Environment Specification Error.

EXCH

Exchange.

Exp

Exponent.

Exponent Aux

Exponent Auxiliary board in ALN.

Ext

External.

Fctn

Function.

FF

Flip-flop.

FIFO

First In/First Out.

FLC

C170 Mode Central Memory Field Length.

60000284 A

IFT3/P - A8

FNO
Fanout.

FRC
Free Running Counter.

FU
Functional Unit.

Gen
Generator.

Gen1
General.

GME
Used by test CTT3/P to indicate a field (bits 62 and 63) within the general micrand.

Hldg
Holding.

I
Operand Register Designator.

IACY
C170 Parcel Select Multiplexer (IFT3/P test).

IAHY
Instruction Assembly Hybrid Multi- plexer (IFT3/P test).

IAPL
C180 Parcel Select Multiplexer (IFT3/P test).

60000284 A IFT3/P - A9

IB02
Instruction Buffer Rank 02 (IFT3/P test).

IB12 valid
Used in ALN test ANT3/P. Instruction Buffer Rank 12 valid signal from Instruction Fetch. This goes active when IF has finished fetching the next instruction from CM.

ICC
Instruction Completion Control.

ICP
Instruction Control Pipeline.

IDS
Immediate Data Select Micrand Field.

IDX
Indexed.

IF
Instruction Fetch.

II
Instruction Issue.

Immed
Immediate.

Incr
Increment.

Indef
Indefinite.

60000284 A IFT3/P - A10

Inf
Infinite.

Inh
Inhibit.

Inp
Input.

Instr
Instruction.

Int
Internal.

Integer Mux
Integer/Exponent Multiplexer in ALN.

Intrpt
Interrupt.

IOU
Input/Output Unit.

ISE
Instruction Specification Error.

IU
Instruction Unit.

J
Operand Register Designator.

60000284 A

IFT3/P - A11

JPS
Job Process State.

k
Operand Register Designator.

K
170 18-Bit Immediate Operand.

K/L
Key/Lock.

Kypt
Keypoint.

l
BDP Operand Length.

Ld
Load.

LM
Local Memory.

LMMIC
Local Memory Functional Micrand field.

LOS
Loss of Significance.

LRR
Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P, and ICT3/P to indicate a live register read.

60000284 A

IFT3/P - A12

LRU
Least Recently Used.

LRW
Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P and ICT3/P to indicate a live register write.

SD
Least Significant Digit.

LSI
Large Scale Integration.

LSM
Last Symbol Mask.

LSMPRE
LOAD/STORE Multiple Address Adder.

Lwr
Lower.

MAC
Maintenance Access Control.

Maint
Maintenance.

MAR
Micrand Address Register.

MBE
Multiple Bit Error.

MC
Master Clear.

MCH
Maintenance Channel.

MCMR
Monitor Condition Mask Register. Also refer to MMR.

MCR
Monitor Condition Register.

Mem
Memory.

Micr
Micrand.

Micrand Constant
Contents of ALN FU Micrand bits 0 through 6.

MIS
Miscellaneous Micrand Field.

Misc
Miscellaneous.

MMR
Monitor Mask Register.

MOP
Micro operation.

MPS
Monitor Process State.

MR
Maintenance Register.

MSB
Most Significant Bit.

MSC
Micrand Sequence Control. A field (bits 0 through 15) within the general micrand.

MSD
Most Significant Digit.

MSNZB
Most Significant Nonzero Byte.

Mult
Multiply.

Mux
Multiplexer.

Neg
Negative.

Norm
Normalize.

NPGK
New P Global Key.

80000284 A

IFT3/P - A15

NPLK
New P Local Key.

ns
Nanosecond.

o
BDP Operand Address.

Op
Operand.

Opcode
Operation Code.

OPGK
Old P Global Key.

OPLK
Old P Local Key.

OPI
Operand Issue.

Opn
Operation.

OSB
Operating System Bounds.

Out
Output.

80000284 A

IFT3/P - A16

Ovf1

Overflow.

P

Program Address.

Par

Parity.

PDM

Processor Detected Malfunction.

PE

Parity Error.

PFA

Page Frame Address.

PFS

Processor Fault Status.

Ph

Phase.

PIT

Process Interval Timer.

PMF

Performance Monitoring Facility.

PN

Page Number.

60000284 A

IFT3/P - A17

PO

Page Offset.

PONR

Point Of No Return.

Pos

Positive.

PP

Peripheral Processor.

Prev

Previous.

Pri

Priority.

Prod

Product.

PSM

Page Size Mask.

PSWF

Page Table Search Without Find.

PTA

Page Table Address.

PTD

Page Table Descriptor.

60000284 A

IFT3/P - A18

PTL
Page Table Length.

PTM
Processor Test Mode.

PVA
Process Virtual Address.

PW
Partial Write.

Q
C180 16-Bit Immediate Operand.

Quad
Quadrant.

Quotient Rgtr
Divide network register internal to the LSI CI arrays.

R/W
Read/Write.

RAC
C170 Mode Central Memory Reference Address.

RAM
Random Access Memory.

RCL
Read and Clear Lock.

80000284 A

IFT3/P - A19

Rcvr
Receiver.

RDS
Register/Data Select.

RDSA
In ALN test ANT3/P, refers to the micrand field that provides the register file address on a write of ALN data and to the data path through DAI for that data.

RDSB
In ALN test ANT3/P, refers to the following data path to ALN. The specified register file is output to the RDSB bus, transferred to the ALN on ADATB, and is passed through the multiple/divide unit and output multiplexer to the general network.

RDSC
In ALN test ANT3/P, refers to following data path. The specified register file is sent to the RDSC bus, transferred to ADATC, then goes to the general network.

Ref
Reference.

Regen
Regenerator.

Rel
Relative.

Rem
Remainder.

Req
Request.

80000284 A

IFT3/P - A20

Resync

Resynchronize, Resynchronizer, Resynchronization.

RF

Register File.

RF0x

Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P to identify the register file being used. x is 5, 6, 7, 8, 9, A, B, C, D, E or F (Hexadecimal).

Rgtr

Register.

RMA

Real Memory Address.

ROM

Read Only Memory.

RSL

Read and Set Lock.

RS64

Right Shift 64 signal from the SKG.

Rxx

Identifies ranks in IF and ICP in Test CTT3/P.

SBE

Single Bit Error.

SC

Soft Control.

60000284 A

IFT3/P - A21

SCM

Soft Control Memory.

SCT

Special Character Table.

SECEDED

Single Error Correction/Double Error Detection.

Seg

Segment.

Sel

Select.

Sep

Separate.

Seq

Sequence.

Shift Mux

Shift Network Multiplexer.

Shifter

Shift Network.

Sig

Significant.

SIT

System Interval Timer.

60000284 A

IFT3/P - A22

SKG

Shift Count Generator.

SM

Segment Map.

SMMIC

Segment Map Functional Micrand Field.

SMO(x)

Segment Descriptor Multiplexer Output bit x.

SMPDM

Segment Map Processor Detected Malfunction.

SN

Segment Number.

Spec

Specification.

SPID

Segment/Page Identifier.

STA

Segment Table Address.

STL

Segment Table Length.

Str

Stream.

80000284 A

IFT3/P - A23

Subtr

Subtract.

Suppr

Suppression.

SV

Specification Value.

SVA

System Virtual Address.

Sw

Switch.

Syn

Syndrome.

Sync

Synchronize.

Termn

Terminate.

TFA

Tag File Address.

ubit

Micrand Bit.

UCEL1

Uncorrected Error Log 1.

80000284 A

IFT3/P - A24

UCEL2

Uncorrected Error Log 2.

UCR

User Condition Register.

UMR

User Mask Register.

Unbr

Unbranch.

Uncond

Unconditional.

Uncor

Uncorrectable.

Undf1

Underflow.

Unlog

Unlogged.

Upr

Upper.

usec

Microsecond.

UTP

Untranslatable Pointer.

60000284 A

IFT3/P - A25

Ux

Used in Multiply/Divide test MDT3/P to identify Major Cycle Control bits 0 through 14.

VMID

Virtual Machine Identifier.

Vx

Used in Multiply/Divide test MDT3/P to identify Minor Cycle Control bits 0 through 15.

DS

Write Data Select.

WOI

Word of Interest.

Xfer

Transfer.

Xltr

Translator.

Xmtr

Transmitter.

XOR

Exclusive OR.

ZIF

Zero Insertion Force.

ZF

Zero Flag.

60000284 A

IFT3/P - A26

IFT3/P TEST MICRANDS

=====

The following is a list of the micrand sequences loaded by the IFT3/P test. The 2-digit number to the left of the sequence corresponds to the control store address of the first micrand in the sequence.

Some of the terms used in this appendix are defined as follows:

Term	Definition
Go to Halt	Go to CSA 10.
Set Halt	Set Halt flag.
Exit	Go to CSA specified by op code 100 through 3FF.
Rev-Date	Microcode revision date.
P-Fetch	Instruction Fetch Program Register.
P-Current	Program address of current virtual instruction being executed.
LRW-P-Fetch	Write P-Fetch.

Unless otherwise specified, the sequence after an exit includes a micrand that loads RF3 with a value equal to the current control store address then branches to control store address 7A(HEX). Control store address 7A(HEX) contains a micrand that will branch to the micrand sequence to be run for the executing subsection.

Sequence Number	Description
00	Go To Here.
02	Set Halt, If Halt Not Set, Go To Here. Else, Go To 03.
03	If Halt Set, Go To Here. Else, Exit.
04	Go To Here.
08	Go To Here.
10	Exit if IB12 is Valid, Else, Go To 10.
30 through 32	Go To Here.
40 through 47	Go To Here.

80000284 A

IFT3/P - B1

Sequence Number	Description
50	Write RF9 with AA(HEX) from FU Micrand, Go To Halt.
51	Initialize IF, Go To Halt.
52	Initialize IF, Write P-Fetch from RF0, Write RF4 with Rev-Date from FU Micrand, Write RF3 with Rev-Date from FU Micrand, Exit.
53	Initialize IF and Clear Stream, Write P-Fetch from RF0, Write RF4 with Rev-Date from FU Micrand Go To Halt.
54	Write RF8 with C180 JKQ from ICP, Write RF4 with P-Current from ICP, Go To Halt.
55	Write RF4 with P-Current from ICP, Go To Halt.
58	Write RF7 with C170 K from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-Current from ICP, Go To Halt.
59	If DAI Condition A, Go To 59-1, Else, Set DAI Condition A, Exit.
59 - 1	Write RF7 with C180 Q from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-Current, Go To Halt.
5C	Initialize IF And Clear Stream, Write RAC from RF1, Write RA+FL from RF2, Go To Halt.
5E	Write VMID Bit from RF10, Go To Halt.

80000284 A

IFT3/P - B2

Sequence Number	Description
60	If DAI Condition A, Go To 60-1. Else, Set DAI Condition A, Write RF5 from RF3, Write RF6 with C180 JKQ from ICP, Exit.
60 - 1	If DAI Condition B, Go To 60-2. Else, Set DAI Conditions A And B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.
60 - 2	If DAI Condition C, Go To 60-3. Else, Set DAI Condition A, B and C, Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
60 - 3	If DAI Condition D, Go To 60-4. Else, Set DAI Conditions A, B, C and D, Write RFB from RF3, Write RFC with C180 JKQ from ICP, Exit.
60 - 4	Write RF4 with P-Current from ICP, Go To Halt.
64	If DAI Condition B, Go To 64-1. Else, Set DAI Conditions A And B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.
64 - 1	If DAI Condition C, Go To 64-2. Else, Set DAI Condition A, B and C Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
64 - 2	If DAI Condition D, Go To 64-3. Else, Set DAI Conditions A, B, C and D, Write RFB from RF3, Write RFC with C180 JKQ from ICP, Exit

Sequence Number	Description
64 - 3	Write RF4 with P-Current from ICP, Go To Halt.
71	If DAI Condition D, Write RF7 with C170K from ICP, Write RF4 with P-Current from ICP, Go To Halt. Else, Set DAI Condition D, Exit.
75	Clear DAI Condition Register, Initialize IF, Write P-Fetch from RF0, Write RF4 with Rev-Date from FU Micrand, Write RF3 with Rev-Date from FU Micrand, Exit.
76	Wait one micrand, Write RF5 from RF3, Write RF7 with C170 K from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-Current from ICP, Go To Halt.
7C	If DAI Condition D, Write RF4 with P-Current from ICP, Go To Halt. Else, Write P-Fetch from RF11, Set DAI Condition D, Exit.
80	Write DAI Condition Register from RF10, Go To Halt.
85	If DAI Condition D, Write RF7 with C170 K from ICP, Write RF8 with C180 JKQ from ICP, Write RF4 with P-Current from ICP, Go To Halt. Else, Set DAI Condition Register, Write P-Fetch from RF13, Exit.
8A	If DAI Condition D is set, Go to 8A-1. Else, write RF04 from RF03. Set DAI Condition D. Exit.

Sequence Number	Description
8A-1	Write RF09 from C170 K instruction portion. Halt.
91	Write RF4 with C180 JKQ from ICP, Write RF3 with P-Current from ICP, Write P-Fetch from RF4, If DAI Condition A, Go To 91-1. Else, Set DAI Condition A, Write RF5 from RF3, Write RF6 with C180 JKQ from ICP, Exit.
91 - 1	If DAI Condition B, Go To 91-2. Else, Set DAI Conditions A And B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.
91 - 2	If DAI Condition C, Go To 91-3. Else, Set DAI Conditions A, B And C, Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
91 - 3	If DAI Condition D, Go To 91-4. Else, Set DAI Conditions A, B, C And D, Write RFB from RF3, Write RFC with C180 JKQ from ICP, Exit.
91 - 4	Write RF4 with P-Current from ICP, Go To Halt.
92	Write RF3 with P-Current from ICP, If DAI Condition A, Go To 92-1. Else, Set DAI Condition A, Write RF5 from RF3, Write RF6 with C180 JKQ from ICP, Exit.
92 - 1	If DAI Condition B, Go To 92-2. Else, Set DAI Condition A And B, Write RF7 from RF3, Write RF8 with C180 JKQ from ICP, Exit.

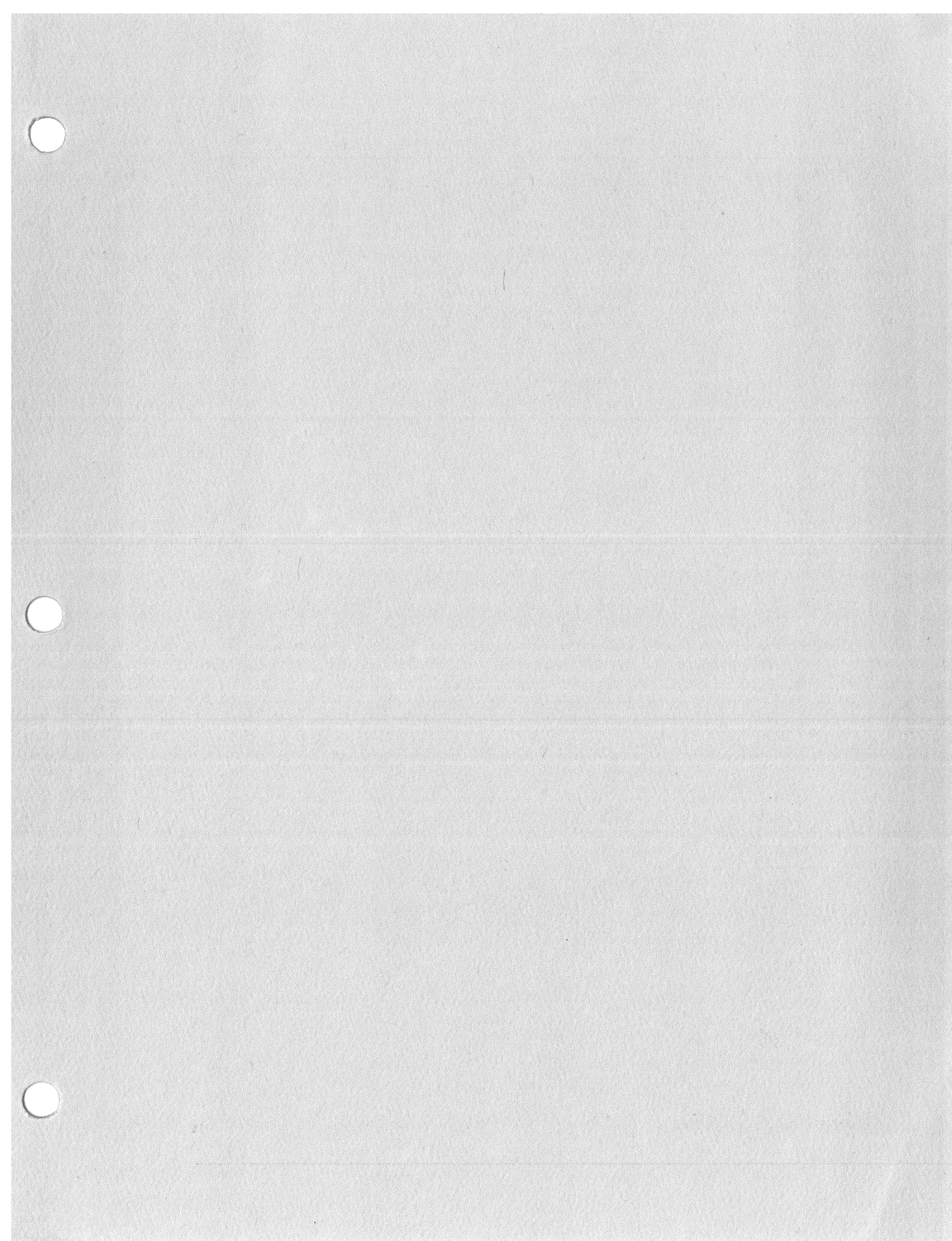
80000284 A

IFT3/P - B5

Sequence Number	Description
92 - 2	If DAI Condition C, Go To 92-3. Else, Set DAI Condition A, B And C, Write RF9 from RF3, Write RFA with C180 JKQ from ICP, Exit.
92 - 3	If DAI Condition D, Go To 92-4. Else, Set DAI Condition A, B, C And D, Write RF8 from RF3, Write RFC with C180 JKQ from ICP, Exit.
92 - 4	Write RF4 with P-Current from ICP, Go To Halt.
95	If 180 Monitor Mode, Go to 95-1 Else, toggle Monitor/Job Mode Flag, Go to 95-1
95 - 1	Write 170 Monitor Flag with 1, Write Trap Enable with zero, Write User Mask with zero, Write Monitor Mask with zero, Write Page Size Mask with zero, Write Error Exit Mask with zero, Go to Halt.

80000284 A

IFT3/P - B6





4201 North Lexington Avenue Saint Paul, Minnesota 55126-6198